



TUGAS AKHIR - KI141502

# ***MINIGAME PUZZLE DAN ARCADE PADA GAME 'AVIAR' BERBASIS VIRTUAL REALITY***

**AUFAR RIZQI**  
**NRP 5114100124**

Dosen Pembimbing I  
Dr.Eng Darlis Herumurti, S.Kom., M.Kom.

Dosen Pembimbing II  
Imam Kuswardayan, S.Kom., M.T.

Departemen Informatika  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018





**TUGAS AKHIR - KI141502**

***MINIGAME PUZZLE DAN ARCADE PADA GAME  
'AVIAR' BERBASIS VIRTUAL REALITY***

**AUFAR RIZQI  
NRP 5114100124**

**Dosen Pembimbing I  
Dr.Eng Darlis Herumurti, S.Kom., M.Kom.**

**Dosen Pembimbing II  
Imam Kuswardayan, S.Kom., M.T.**

**Departemen Informatika  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018**

***(Halaman ini sengaja dikosongkan)***



**UNDERGRADUATE THESES - KI141502**

# **PUZZLE AND ARCADE MINIGAME ON ‘AVIAR’, A VIRTUAL REALITY BASED GAME**

**AUFAR RIZQI**  
**NRP 5114100124**

**First Advisor**  
**Dr. Darlis Herumurti, S.Kom., M.Kom.**

**Second Advisor**  
**Imam Kuswardayan, S.Kom., M.T.**

**Department of Informatics**  
**Faculty of Information Technology and Communication**  
**Sepuluh Nopember Institute of Technology**  
**Surabaya 2018**

***(Halaman ini sengaja dikosongkan)***

## LEMBAR PENGESAHAN

### **MINIGAME PUZZLE DAN ARCADE PADA GAME 'AVIAR' BERBASIS VIRTUAL REALITY**

#### **TUGAS AKHIR**

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Bidang Studi Interaksi Grafika dan Seni  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**AUFAR RIZQI**  
**NRP: 5114100124**

Disetujui oleh Pembimbing Tugas Akhir:

1. Dr.Eng Darlis Herumurti, S.Kom., M.Kom. ....  
(NIP. 197712172003121001) (Pembimbing 1)
2. Imam Kuswardayan, S.Kom., M.T. ....  
(NIP. 197612152003121001) (Pembimbing 2)



**SURABAYA**  
**JULI, 2018**

***(Halaman ini sengaja dikosongkan)***



## **MINIGAME PUZZLE DAN ARCADE PADA GAME 'AVIAR' BERBASIS VIRTUAL REALITY**

**Nama Mahasiswa : AUFAR RIZQI**  
**NRP : 511400124**  
**Jurusan : Teknik Informatika FTIK-ITS**  
**Dosen Pembimbing 1 : Dr.Eng Darlis Herumurti, S.Kom.,  
M.Kom.**  
**Dosen Pembimbing 2 : Imam Kuswardayan, S.Kom., M.T.**

### **ABSTRAK**

*Tidak bisa dipungkiri bahwa teknologi dari tahun ke tahun selalu berkembang. Salah satu yang sedang marak sekarang adalah Virtual Reality. Teknologi Virtual Reality atau yang sering disebut dengan VR merupakan teknologi yang bisa membuat pengguna masuk ke dalam dunia virtual dan merasakan langsung dunia di dalamnya.*

*Ide yang digunakan dalam tugas akhir ini adalah membangun sebuah game berbasis virtual reality dimana pemain akan ditempatkan di dalam sebuah labirin. Lalu tujuan dari permainan ini adalah untuk keluar dari labirin tersebut. Namun labirin tersebut akan berubah pola dalam kurun waktu tertentu. Selain itu, akan ada musuh yang mencoba menghadang pemain dalam melarikan diri. Untuk membantu pemain melarikan diri dari labirin, tersedia minigame. Ketika menyelesaikan sebuah minigame, maka pemain akan mendapat barang bantuan berupa senjata untuk melawan musuh yang ada.*

*Berdasarkan hasil pengujian dihasilkan permainan labirin yang dapat berubah bentuk sesuai dengan waktu siang dan malam, permainan dapat berjalan pada Oculus Rift, dibangun dengan Unity versi 2017.3.0f3 dengan bahasa pemrograman C#, asset yang digunakan sebagian besar diambil dari internet atau dibuat di dalam Unity. Dengan pengujian beta dan dari kuesioner dapat*

*disimpulkan permainan telah mengimplementasikan perancangan dengan baik.*

***Kata kunci: Virtual Realiy, Labirin***

## **PUZZLE AND ARCADE MINIGAME ON ‘AVIAR’, A VIRTUAL REALITY BASED GAME**

**Student’s Name** : AUFAR RIZQI  
**Student’s ID** : 5114100124  
**Department** : Teknik Informatika FTIK-ITS  
**First Advisor** : Dr.Eng Darlis Heru Murti, S.Kom.,  
M.Kom.  
**Second Advisor** : Imam Kuswardayan, S.Kom., M.T.

### **ABSTRACT**

*We can’t avoid from the reality that technology from time to time are evolving. One of them that is very popular right now is Virtual Reality. The technology of virtual reality or as we call it VR is a technology that put user into a virtual world and experience it. The use of VR are simulation and game.*

*The idea used in this final project is to build a virtual reality-based game where player will be put inside a maze. The objective is to get out of the maze. Yet the maze will be randomized every period of time. Other than that there will be enemy that prevent the player from escaping the maze. To help player escape, there are minigames to play. After player finish a minigame, player will get a weapon to fight the enemy.*

*According to testing that result a maze game that can change pattern according to the day and night cycle, this game can run on Oculus Rift, built using Unity version 2017.3.0f3 with C# language programming, the assets that are used inside the game are mostly taken from the internet or made directly in Unity, with beta testing and questionnaire writer can conclude that the game has implemented the design well.*

***Keywords : Virtual Reality, Maze***

***(Halaman ini sengaja dikosongkan)***

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul:

### ***“MINIGAME PUZZLE DAN ARCADE PADA GAME ‘AVIAR’ BERBASIS VIRTUAL REALITY”***

Terselesaikannya Tugas Akhir ini tidak terlepas dari bantuan dan dukungan banyak pihak, Oleh karena itu melalui lembar ini penulis ingin mengucapkan terima kasih dan penghormatan kepada:

1. Allah SWT serta junjungan Nabi Muhammad SAW, karena limpahan rahmat dan karunia-Nya penulis dapat menyelesaikan Tugas Akhir dan juga perkuliahan di Teknik Informatika ITS.
2. Ayah dan Ibu penulis, Ismunandar dan Nuning yang tiada hentinya memberikan dukungan doa, moral, dan material kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini.
3. Kakak penulis, Annisha Nurilla dan keluarga tercinta yang telah memberikan dukungan dan semangatnya kepada penulis.
4. Bapak Dr.Eng Darlis Herumurti, S.Kom., M.Kom. dan Bapak Imam Kuswardayan, S.Kom., M.T. selaku pembimbing I dan II yang telah membimbing dan memberikan motivasi, nasihat dan bimbingan dalam menyelesaikan Tugas Akhir ini.
5. Bapak Dr.Eng Darlis Herumurti, S.Kom., M.Sc. selaku dosen wali penulis yang telah memberikan arahan kepada

penulis selama menjalani perkuliahan di Teknik Informatika ITS.

6. Bapak Darlis Herumurti, S.Kom., M.Kom. selaku kepala jurusan Teknik Informatika ITS dan segenap dosen dan karyawan Teknik Informatika ITS yang telah memberikan ilmu dan pengalaman kepada penulis selama menjalani masa studi di Teknik Informatika ITS.
7. Kawan satu kelompok tugas akhir, Anggit Yudhistira dan Vinsensia Sipriana Z yang telah memberikan masukan dan semangat ketika penulis sedang kehilangan arah.
8. Sahabat penulis Nafia, Ikhsan, Aldo, Resha, Bimo, Deni, Dwika, Elva, Valdy, Rifat, Shafly (Om), Faishal, Tosca, Rara, Ghaly dan teman seperjuangan penulis lainnya, teman-teman TC 2013, kakak-kakak TC 2012 & 2013 dan adik-adik TC 2015 & 2016 yang tidak dapat disebutkan satu per satu yang selalu membantu, menghibur, menjadi tempat bertukar ilmu serta pembelajaran baru dan berjuang bersama-sama penulis.
9. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa laporan Tugas Akhir ini masih memiliki banyak kekurangan. Oleh karena itu dengan segala kerendahan hati penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan penulis kedepannya. Selain itu, penulis berharap laporan Tugas Akhir ini dapat berguna bagi pembaca secara umum.

Surabaya, Mei 2018

## DAFTAR ISI

<b>LEMBAR PENGESAHAN.....</b>	<b>v</b>
<b>ABSTRAK.....</b>	<b>vii</b>
<b>ABSTRACT .....</b>	<b>ix</b>
<b>KATA PENGANTAR .....</b>	<b>xi</b>
<b>DAFTAR ISI.....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR .....</b>	<b>xvii</b>
<b>DAFTAR TABEL.....</b>	<b>xix</b>
<b>DAFTAR KODE SUMBER .....</b>	<b>xxi</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Permasalahan .....	2
1.4 Tujuan .....	3
1.5 Manfaat.....	3
1.6 Metodologi .....	3
1.6.1 Penyusunan Proposal Tugas Akhir .....	3
1.6.2 Studi Literatur .....	4
1.6.3 Implementasi Perangkat Lunak.....	4
1.6.4 Pengujian dan Evaluasi.....	4
1.6.5 Penyusunan Buku .....	4
1.7 Sistematika Penulisan.....	4
<b>BAB II DASAR TEORI.....</b>	<b>7</b>
2.1 Oculus Rift .....	7
2.2 Virtual Reality .....	14
2.3 <i>Human Computer Interaction</i> .....	14
2.4 Unity (Game Engine) .....	15
2.5 Pemodelan Tiga Dimensi (3D).....	15
2.6 <i>Visual Studio</i> .....	15
2.7 Blender .....	16

2.8	Whac-A-Mole.....	17
2.9	Mastermind.....	17
<b>BAB III PERANCANGAN SISTEM.....</b>		<b>19</b>
3.1	Deskripsi Umum Perangkat Lunak.....	19
3.2	Arsitektur Permainan.....	20
3.3	Perancangan Skenario Permainan .....	21
3.3.1	Kontroler.....	21
3.3.2	Legot.....	23
3.3.3	Picture Memory .....	25
3.3.4	Tap Fast.....	27
3.4	Perancangan Antarmuka.....	30
3.4.1	Legot.....	30
3.4.2	Picture Memory .....	31
3.4.3	Tap Fast.....	32
<b>BAB IV IMPLEMENTASI.....</b>		<b>35</b>
4.1	Lingkungan Implementasi .....	35
4.2	Implementasi pada Permainan.....	35
4.2.1	Penempatan <i>Minigame</i> Pada Labirin .....	36
4.2.2	Kontroler.....	44
4.2.3	Legot.....	47
4.2.3.1	Legot.....	47
4.2.3.2	Data Controller .....	50
4.2.3.3	Legot Button.....	52
4.2.3.4	Soal Legot.....	56
4.2.3.5	Jawaban Player .....	57
4.2.4	Picture Memory .....	60
4.2.4.1	Picture Memory .....	60
4.2.4.2	Picture Buttons .....	64
4.2.4.3	Picture Button.....	68
4.2.5	Tap Fast.....	72
4.2.5.1	Tap Fast .....	72
4.2.5.2	Start Button.....	76
4.2.5.3	Button .....	77
4.2.5.4	Tap Fast Text.....	80



4.2.6	Chest .....	81
<b>BAB V</b>	<b>UJI COBA DAN EVALUASI.....</b>	<b>83</b>
5.1	Lingkungan Uji Coba .....	83
5.1.1	Pengujian Fungsionalitas .....	83
5.1.2	Uji Coba Kontroler .....	84
5.1.3	Uji Coba <i>Minigame</i> Legot .....	85
5.1.4	Uji Coba <i>Minigame</i> Picture Memory .....	89
5.1.5	Uji Coba <i>Minigame</i> Tap Fast .....	92
5.1.6	Hasil Pengujian .....	95
5.2	Pengujian Pengguna .....	96
5.2.1	Skenario Pengujian Pengguna .....	96
5.2.2	Daftar Penguji .....	100
5.2.3	Hasil Pengujian Pengguna .....	100
5.2.4	Kritik dan Saran Pengguna .....	103
5.3	Evaluasi Pengujian .....	104
<b>BAB VI</b>	<b>KESIMPULAN DAN SARAN .....</b>	<b>107</b>
6.1	Kesimpulan.....	107
6.2	Saran.....	107
<b>DAFTAR PUSTAKA .....</b>		<b>109</b>
<b>LAMPIRAN.....</b>		<b>111</b>
<b>BIODATA PENULIS.....</b>		<b>125</b>

*(Halaman ini sengaja dikosongkan)*

## DAFTAR GAMBAR

Gambar 2.1 Oculus headset.....	8
Gambar 2.2 Oculus touch.....	8
Gambar 2.3 Oculus Touch pada dunia nyata dan virtual .....	9
Gambar 2.4 Oculus remote.....	9
Gambar 2.5 Joystick Xbox .....	10
Gambar 2.6 Connector VR Guitar.....	11
Gambar 2.7 VR Guitar Controller, Oculus Touch, dan Connector VR Guitar .....	11
Gambar 2.8 Cara menghubungkan Oculus Touch dengan VR Guitar Controller .....	12
Gambar 2.9 Jangkauan deteksi Oculus Sensor pada ruangan .....	13
Gambar 2.10 Oculus sensor.....	13
Gambar 2.11 Prinsip kerja komputer.....	14
Gambar 2.12 Membuat model 3D pada blender .....	16
Gambar 3.1 Arsitektur <i>Game AVIAR</i> .....	20
Gambar 3.2 Laser pada oculus touch .....	22
Gambar 3.3 Tombol-tombol pada oculus touch .....	22
Gambar 3.4 Flowchart <i>minigame</i> Legot.....	24
Gambar 3.5 Flowchart <i>minigame</i> Picture Memory .....	26
Gambar 3.6 Flowchart <i>minigame</i> Tap Fast .....	29
Gambar 3.7 Antarmuka <i>minigame</i> Legot .....	30
Gambar 3.8 Antarmuka <i>minigame</i> Picture Memory .....	31
Gambar 3.9 Antarmuka <i>minigame</i> Tap Fast.....	32
Gambar 4.1 Implementasi <i>minigame</i> Legot .....	47
Gambar 4.2 Implementasi Picture Memory .....	60
Gambar 4.3 Implementasi Tap Fast .....	72
Gambar 5.1 Hasil uji coba UF-001 Skenario 1 .....	85
Gambar 5.2 Hasil uji coba UF-001 Skenario 2 .....	85
Gambar 5.3 Hasil uji coba UF-002 Skenario 1 .....	87
Gambar 5.4 Hasil uji coba UF-002 Skenario 2 .....	87
Gambar 5.5 Hasil uji coba UF-002 Skenario 3 .....	88
Gambar 5.6 Hasil uji coba UF-002 Skenario 4 .....	88
Gambar 5.7 Hasil uji coba UF-003 Skenario 1 .....	90

Gambar 5.8 Hasil uji coba UF-003 Skenario 2 .....	91
Gambar 5.9 Hasil uji coba UF-003 Skenario 3 .....	91
Gambar 5.10 Hasil uji coba UF-003 Skenario 4 .....	92
Gambar 5.11 Hasil uji coba UF-004 Skenario 1 .....	94
Gambar 5.12 Hasil uji coba UF-004 Skenario 2.....	94
Gambar 5.13 Hasil uji coba UF-004 Skenario 3.....	95

## DAFTAR TABEL

Tabel 3.1 Kebutuhan fungsionalitas Kontroler .....	23
Tabel 3.2 Kebutuhan fungsionalitas <i>minigame</i> Legot.....	25
Tabel 3.3 Kebutuhan fungsionalitas <i>minigame</i> Picture Memory .....	27
Tabel 3.4 Pembagian tingkat kesulitan <i>minigame</i> Tap Fast .....	28
Tabel 3.5 Kebutuhan fungsionalitas <i>minigame</i> Tap Fast .....	29
Tabel 4.1 Spesifikasi lingkungan implementasi.....	35
Tabel 5.1 Spesifikasi lingkungan uji coba.....	83
Tabel 5.2 Skenario Uji Coba Kontroler.....	84
Tabel 5.3 Skenario Uji Coba <i>Minigame</i> Legot.....	85
Tabel 5.4 Skenario Uji Coba <i>Minigame</i> Picture Memory .....	89
Tabel 5.5 Skenario Uji Coba <i>Minigame</i> Tap Fast .....	92
Tabel 5.6 Hasil Pengujian Fungsionalitas .....	95
Tabel 5.7 Pertanyaan Karakteristik Pengguna.....	97
Tabel 5.8 Rentang Nilai.....	97
Tabel 5.9 Pertanyaan Penilaian <i>Minigame</i> .....	97
Tabel 5.10 Daftar Nama Pengguna .....	100
Tabel 5.11 Hasil Pengujian Pengguna.....	100
Tabel 5.12 Hasil Akhir Pengujian Pengguna .....	102
Tabel 5.13 Kritik dan Saran Pengguna.....	104

*(Halaman ini sengaja dikosongkan)*

## **DAFTAR KODE SUMBER**

Kode Sumber 4.1 Kelas MinigameSpawner .....	44
Kode Sumber 4.2 Kelas MinigameInteraction .....	46
Kode Sumber 4.3 Kelas LegotGameController.....	50
Kode Sumber 4.4 Kelas LegotDataContainer .....	52
Kode Sumber 4.5 Kelas LegotInput .....	54
Kode Sumber 4.6 Kelas LegotRestartAnswer.....	56
Kode Sumber 4.7 Kelas LoadSoal .....	57
Kode Sumber 4.8 Kelas PrintKombinasiJawaban.....	58
Kode Sumber 4.9 Kelas PrintJawabanPlayer .....	59
Kode Sumber 4.10 Kelas PictureMemoryMaster.....	64
Kode Sumber 4.11 Kelas PictureMemoryButtonPos .....	68
Kode Sumber 4.12 Kelas PictureMemoryController .....	71
Kode Sumber 4.13 Kelas TapFastController.....	76
Kode Sumber 4.14 Kelas TapFastStart .....	77
Kode Sumber 4.15 Kelas TapFastButton .....	80
Kode Sumber 4.16 Kelas TapFastText.....	81
Kode Sumber 4.17 Kelas ChestController .....	82

***(Halaman ini sengaja dikosongkan)***



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Teknologi virtual reality (VR) merupakan salah satu teknologi yang sedang banyak dikembangkan. VR sendiri adalah sebuah teknologi dimana pengguna akan ditempatkan pada sebuah dunia virtual yang di dalamnya, pengguna dapat merasakan langsung apa yang terjadi di dalam dunia tersebut. Virtual reality telah banyak dikembangkan di berbagai bidang, seperti militer, kesehatan, pendidikan, hiburan, film, olahraga, bisnis, konstruksi, dan lain-lain. Sistem seperti ini biasanya dapat digunakan untuk perancangan obat, arsitek, insinyur, pekerja medis, latihan perang, bahkan orang awam untuk melakukan kegiatan yang menuri dunia nyata.

Virtual reality pertama kali dijual secara luas dan komersil pada tahun 1990 oleh perusahaan Sega dengan produknya Sega VR. Sega VR merupakan HMD (Head Mounted Display) yang dilengkapi oleh layar LCD, headphone, dan sensor inersia untuk mendeteksi pergerakan kepala dari penggunanya. Kemudian banyak yang mengembangkan teknologi sejenis Sega VR yang sekarang kita kenal dua alat VR yang paling terkenal yaitu Oculus Rift dan HTC Vive.

Secara garis besar, virtual reality mengenalkan kepada kita sebuah pengalaman digital yang imersif dan nyata dari teknologi – teknologi sebelumnya. Dengan virtual reality kita bisa merasakan sesuatu tanpa harus langsung terjun di dunia nyata dan kita bisa menghindari resiko yang mungkin terjadi di dunia nyata. Hal ini yang membuat teknologi VR banyak digunakan dalam simulasi.

Selain dalam simulasi, banyak juga permainan / *game* berbasis VR. Hal ini banyak dilakukan agar pemain bisa merasakan langsung dunia yang ada dalam permainan tersebut. Selain bisa berinteraksi dengan lingkungan yang ada dan pemain juga dapat mengeksplorasi dunia di dalamnya. Hal ini memberi sebuah pengalaman yang berbeda dan baru bagi semua orang karena

selama ini kita hanya bisa memainkan *game* melalui layar saja. Pemain bisa merasakan langsung ketegangan dan keseruan yang ada. Salah satu permainan yang mengasah kemampuan spasial dan memori adalah permainan labirin.

Labirin merupakan sebuah *game puzzle* berbentuk kumpulan jalan yang memiliki titik awal / titik masuk dan titik akhir / titik keluar. Tujuan dari permainan ini adalah mencari jalan untuk mencapai titik akhir. Untuk mencari jalan keluar dari labirin, seseorang harus bisa mengingat jalan yang ia ambil dan membayangkan dirinya dalam labirin tersebut agar tahu posisinya dalam labirin. Labirin sendiri masih terhitung jarang bisa ditemukan karena membutuhkan biaya yang besar dan lahan yang luas untuk membuat sebuah labirin.

Dengan menggabungkan dua elemen yaitu Virtual Reality dan labirin, penulis membuat sebuah *game* labirin berbasis *virtual reality*.

## 1.2 Rumusan Masalah

Tugas Akhir ini mengangkat beberapa rumusan masalah sebagai berikut:

1. Bagaimana mengimplementasikan *minigame* pada permainan 'AVIAR'?
2. Bagaimana merancang permainan yang bisa melatih pemain dalam hal tertentu?
3. Bagaimana merancang skenario dan tingkat kesulitan *minigame* pada permainan 'AVIAR'?

## 1.3 Batasan Permasalahan

Permasalahan yang dibahas pada Tugas Akhir ini memiliki batasan sebagai berikut:

1. Permainan dapat dimainkan maksimal oleh dua pemain.
2. Aplikasi yang dibangun berjalan pada platform Oculus Rift.

3. Lingkungan pengembangan yang digunakan menggunakan aplikasi Unity 3D Free License dan bahasa pemrograman C#.

## **1.4 Tujuan**

Tujuan dari pembuatan Tugas Akhir ini adalah sebagai berikut:

1. Membuat *minigame* yang dapat diimplementasikan pada sebuah permainan labirin.
2. Membuat permainan yang tidak hanya seru dimainkan tetapi juga bermanfaat.
3. Membuat *minigame* dengan tingkat kesulitan yang berbeda-beda.

## **1.5 Manfaat**

Manfaat dari Tugas Akhir ini adalah sebagai pengembangan Oculus Rift sebagai alat dalam permainan yang bisa memberikan pengalaman baru bagi pemain. Selain itu pemain juga dapat merasakan keseruan bermain labirin yang jumlahnya tidak banyak di dunia nyata.

## **1.6 Metodologi**

Pembuatan Tugas Akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

### **1.6.1 Penyusunan Proposal Tugas Akhir**

Tahapan awal dari Tugas Akhir ini adalah penyusunan Proposal Tugas Akhir. Proposal Tugas Akhir berisi pendahuluan, deskripsi dan gagasan metode – metode yang dibuat dalam Tugas Akhir ini. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya Tugas Akhir, rumusan masalah yang diangkat, batasan masalah untuk Tugas Akhir, dan manfaat dari hasil pembuatan Tugas Akhir ini. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan Tugas Akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan Tugas Akhir.

### **1.6.2 Studi Literatur**

Dalam studi literatur ini penulis akan mempelajari beberapa referensi terkait topik Tugas Akhir. Beberapa referensi tersebut adalah mengenai Oculus Rift Controller, Oculus Rift SDK, dan Unity.

### **1.6.3 Implementasi Perangkat Lunak**

Aplikasi yang akan dibuat ini dibangun menggunakan Game Engine Unity 3D Free, dengan Bahasa Pemrograman C#. Aplikasi ini juga menggunakan Oculus Rift SDK yang akan dihubungkan dengan Oculus Rift Controller.

### **1.6.4 Pengujian dan Evaluasi**

Tahap Pengujian dan Evaluasi berisi pengujian aplikasi dan evaluasi berdasarkan hasil pengujian. Pengujian akan dilakukan oleh sampel pengguna, yaitu mahasiswa Teknik Informatika ITS sebagai Sampel dari masyarakat Indonesia pada umumnya.

### **1.6.5 Penyusunan Buku**

Pada tahap ini dilakukan penyusunan buku yang menjelaskan seluruh konsep, teori dasar dari metode yang digunakan, implementasi, serta hasil yang telah dikerjakan sebagai dokumentasi dari pelaksanaan Tugas Akhir.

## **1.7 Sistematika Penulisan**

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

### **Bab I Pendahuluan**

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan

masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

## **Bab II Dasar Teori**

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

## **Bab III Analisis dan Perancangan Sistem**

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka pada aplikasi.

## **Bab IV Implementasi**

Bab ini berisi implementasi dari perancangan perangkat lunak.

## **Bab V Pengujian dan Evaluasi**

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat keluaran yang dihasilkan oleh aplikasi dan evaluasi untuk mengetahui kemampuan aplikasi serta mengetahui penilaian aspek kegunaan (*usability*) dari perangkat lunak.

## **Bab VI Kesimpulan dan Saran**

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

## **Daftar Pustaka**

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

## **Lampiran**

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

*(Halaman ini sengaja dikosongkan)*

## **BAB II**

### **DASAR TEORI**

Bab ini berisi pembahasan mengenai teori-teori dasar yang digunakan dalam Tugas Akhir. Teori-teori tersebut diantaranya adalah Oculus Rift Controller, *Human Computer Interaction*, dan beberapa teori lain yang mendukung pembuatan Tugas Akhir.

#### **2.1 Oculus Rift**

Oculus Rift merupakan sebuah kumpulan perangkat virtual reality yang dikembangkan oleh perusahaan Oculus. Pada tahun 2012, Oculus memulai sebuah kampanye kickstarter. Proyek tersebut menuai kesuksesan dengan mengumpulkan dana sebesar 2,5 juta dollar Amerika [1]. Kemudian pada bulan Maret 2014, Facebook membeli perusahaan Oculus sebesar 2 milyar dollar Amerika. Namun pada tahun 2016, pendiri Oculus memutuskan untuk meninggalkan Facebook.

Oculus Rift terdiri atas beberapa komponen, yaitu Oculus Headset, Oculus Touch, Oculus Sensor, Oculus Remote, Joystick Xbox, dan Connector VR Guitar Controller [2].

Oculus Headset merupakan HMD yang dikembangkan oleh Oculus. Alat ini dilengkapi dengan layar LCD untuk menampilkan gambar dan headphone untuk sumber suara. Bentuk dari oculus headset sendiri bisa dilihat pada Gambar 2.1



**Gambar 2.1 Oculus headset**

(Sumber: <https://www.amazon.co.uk/Oculus-301-00204-01-Rift/dp/B00ZFOGHRG>)

Oculus memiliki empat jenis kontroler, yaitu Oculus Touch, Oculus Remote, Joystick Xbox, dan VR Guitar Controller. Beberapa aplikasi bisa menggunakan lebih dari satu jenis controller. Oculus Touch memiliki dua kontroler untuk masing – masing tangan yang bisa dilihat pada Gambar 2.2. Cara menggunakannya adalah dengan memegangnya pada kedua tangan seperti pada Gambar 2.3. Controller ini akan menjadi tangan kita di dalam dunia VR.



**Gambar 2.2 Oculus touch**

(Sumber: <https://www.oculus.com/rift/accessories/>)





**Gambar 2.3 Oculus Touch pada dunia nyata dan virtual**

(Sumber: <https://www.electronicweekly.com/blogs/gadget-master/consumer-electronics/tearing-oculus-rift-touch-controller-2017-01/>)

Lalu ada juga Oculus Remote yang bisa dilihat pada Gambar 2.4. Alat ini berfungsi mirip dengan remote tv. Alat ini tidak bisa berfungsi seperti tangan seperti oculus touch. Namun bisa juga digunakan di dalam game / aplikasi untuk memilih opsi – opsi yang ada.



**Gambar 2.4 Oculus remote**

(Sumber: <https://www.oculus.com/rift/accessories/>)

Selain itu ada juga Joystick Xbox yang berfungsi persis seperti joystick pada console lainnya seperti pada Gambar 2.5. Namun joystick ini dapat digunakan di dalam game / aplikasi VR.



**Gambar 2.5 Joystick Xbox**

(Sumber: <https://www.xbox.com/en-US/xbox-one/accessories/controllers/xbox-black-wireless-controller>)

Dan yang terakhir adalah VR Guitar Controller yang bisa dilihat pada Gambar 2.6. Alat ini berbentuk seperti gitar pada umumnya, namun pada bagian leher gitar terdapat tombol – tombol yang digunakan sebagai input di dalam game. Untuk menyambungkan controller gitar dan oculus, digunakan konektor seperti pada Gambar 2.7. Konektor tersebut akan menghubungkan salah satu bagian oculus touch dengan kontroler gitar seperti pada Gambar 2.8.



**Gambar 2.7 VR Guitar Controller, Oculus Touch, dan Connector VR Guitar**

(Sumber: <https://www.vrheads.com/how-connect-your-oculus-touch-controller-rock-band-guitar>)



**Gambar 2.6 Connector VR Guitar**

(Sumber: <https://www.oculus.com/rift/accessories/>)



**Gambar 2.8 Cara menghubungkan Oculus Touch dengan VR Guitar Controller**

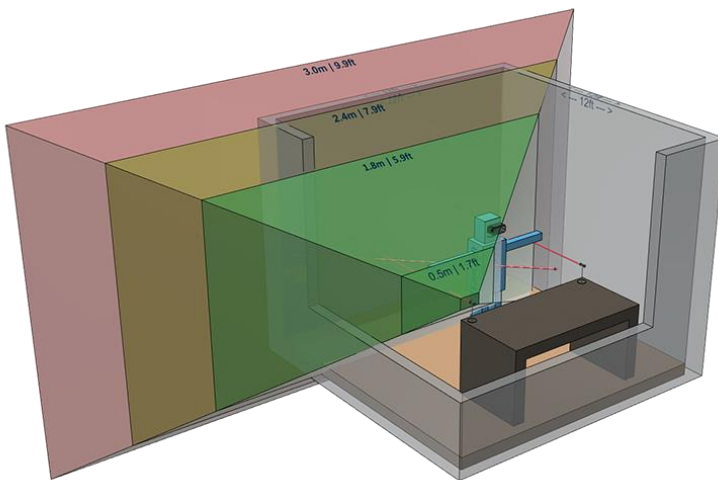
(Sumber: <https://www.tomshardware.com/news/rock-band-vr-oculus-touch,32781.html>)

Selain kontroler, oculus rift juga dilengkapi dengan sensor untuk mendeteksi pergerakan kontroler. Oculus Sensor dapat dilihat di Gambar 2.9. Hal ini dibutuhkan untuk mengetahui pergerakan dari pemain agar bisa diproyeksikan ke dalam dunia virtual. Oculus Sensor sendiri memiliki cakupan deteksi yang lumayan besar, hal tersebut bisa digambarkan pada Gambar 2.10.



**Gambar 2.10 Oculus sensor**

(Sumber: <https://www.oculus.com/rift/accessories/>)



**Gambar 2.9 Jangkauan deteksi Oculus Sensor pada ruangan**

(Sumber: <https://www.roadtovr.com/oculus-roomscale-advanced-setup-tips/>)

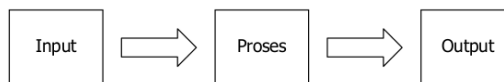
## 2.2 Virtual Reality

Virtual Reality (VR) merupakan sebuah sebutan yang populer dari sebuah dunia maya bersifat interaktif dimana pengguna ditempatkan pada dunia yang disimulasikan dengan menggunakan alat perantara dalam melakukan interaksi manusia dan komputer [3]. VR memungkinkan penggunanya melakukan interaksi dengan dunia maya secara langsung menggunakan alat khusus.

## 2.3 Human Computer Interaction

*Human Computer Interaction*, atau dalam bahasa Indonesia yaitu “Interaksi Manusia dan Komputer” merupakan ilmu yang mempelajari tentang hubungan antara manusia yang pada umumnya digunakan untuk mengendalikan, berkomunikasi, dan memasukkan data pada komputer [4].

Hubungan antara manusia dan komputer tersebut mempunyai karakteristik tertentu untuk mencapai suatu tujuan tertentu dengan menjalankan sebuah sistem yaitu *interface* (antarmuka)



**Gambar 2.11 Prinsip kerja komputer**

Pada gambar 2.11, digambarkan prinsip kerja komputer dimana kepada komputer diberikan input oleh pengguna. Kemudian diolah di dalam komputer yang hasilnya menjadi output sesuai dengan keinginan pengguna.

Jika kita perhatikan dari kasus diatas, maka bisa kita ketahui bahwa pengguna telah berinteraksi dengan komputer. Hal ini bisa kita lihat dari kegiatan yang dilakukan pengguna dan komputer. Pertama pengguna menekan tombol pada keyboard atau mouse. Dengan ditekannya tombol, maka perangkat tersebut

mengirimkan informasi atau data kepada komputer untuk diproses. Setelah diproses, komputer mengirimkan kembali informasi yang diinginkan oleh pengguna. Informasi tersebut ditampilkan oleh komputer pada layar atau speaker yang kita sebut sebagai output.

## **2.4 Unity (Game Engine)**

Unity merupakan sebuah *game engine* yang dikembangkan oleh Unity Technologies. Unity dapat menciptakan *game* ke dalam beberapa sistem operasi sekaligus. Antara lain: Windows Phone, Android, iOS, Windows 8, OSX, Blackberry 10, Playstation 3, Playstation 4, XBOX, dan sebagainya. *Game* yang dapat dibuat dengan Unity ini bisa dalam bentuk 3D atau 2D, tergantung pada developer *game* tersebut. Unity mampu mengubah gambar statis menjadi animasi yang dapat dimainkan. Karena itu lah banyak partner yang menggunakan Unity, antara lain: Microsoft, Sony, Qualcomm, Blackberry, Samsung, dan Nintendo [5].

## **2.5 Pemodelan Tiga Dimensi (3D)**

Pemodelan adalah membentuk suatu benda-benda atau objek. Membuat dan mendesain objek tersebut sehingga terlihat seperti hidup. Sesuai dengan objek dan basisnya, proses ini secara keseluruhan dikerjakan di komputer. Melalui konsep dan proses desain, keseluruhan objek bisa diperlihatkan secara 3 dimensi, sehingga banyak yang menyebut hasil ini sebagai pemodelan 3 dimensi [6].

Pemodelan 3 dimensi memiliki beberapa aspek yang harus diperhatikan, yaitu pendeskripsian objek, tujuan dari model, tingkat kerumitan, kesesuaian dan kenyamanan, serta kemudahan manipulasi objek. Pemodelan 3 dimensi dapat dilakukan dengan bantuan aplikasi-aplikasi, seperti Blender, K-3D, Google SketchUp, dan masih banyak lainnya.

## **2.6 Visual Studio**

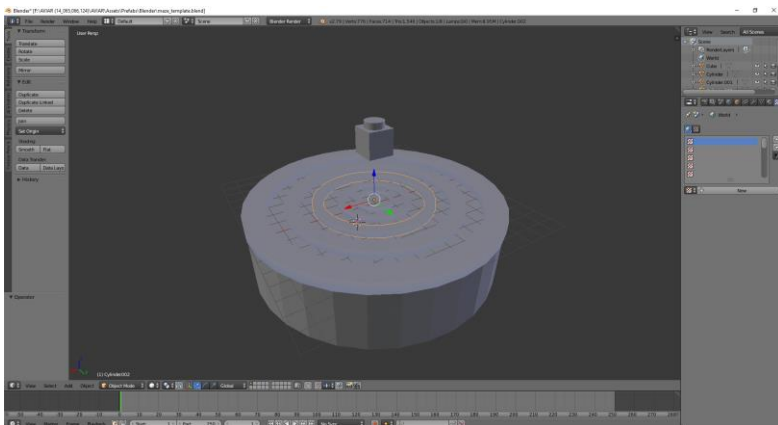
Microsoft Visual Studio merupakan sebuah aplikasi yang terdiri dari kompiler, SDK, Integrated Development Environment

(IDE), dan dokumentasi yang biasa digunakan untuk mengembangkan aplikasi personal, aplikasi web, aplikasi mobile, aplikasi konsol maupun komponen aplikasi pada Microsoft Visual Studio. Banyak pengembang aplikasi yang mengembangkan aplikasinya menggunakan Microsoft Visual Studio karena banyak fungsi yang telah disediakan oleh Microsoft Visual Studio, sehingga pengembang lebih mudah dan sangat terbantu dalam mengembangkan maupun membuat aplikasi [7].

## 2.7 Blender

Blender adalah sebuah perangkat lunak yang berfungsi untuk permodelan 3 dimensi. Blender juga dapat digunakan untuk membuat animasi 3 dimensi. Blender dapat diunduh pada situs resminya tanpa dikenakan biaya. Blender dapat digunakan pada berbagai jenis sistem operasi. Keluaran dari Blender adalah objek-objek 3 dimensi dengan berbagai format, seperti .obj, .3ds, dan lain-lain [8].

Objek keluaran dari Blender dapat digunakan sebagai material dasar pembuatan aplikasi atau permainan dengan menggunakan game engine atau perangkat lunak pembuat aplikasi 3 dimensi. Contoh hasil pemodelan 3D dengan Blender dapat dilihat pada Gambar 2.12.



**Gambar 2.12 Membuat model 3D pada blender**



## 2.8 Whac-A-Mole

Whac-A-Mole merupakan permainan *arcade* yang diciptakan pada tahun 1975 oleh Kazuo Yamada yang kemudian dilisensi oleh Bandai pada tahun 1977 [9]. Permainan ini terdiri dari papan dengan lubang. Dari lubang-lubang tersebut tikus tanah akan keluar. Tugas dari pemain adalah memukul tikus tanah dengan palu untuk mendapatkan poin.

## 2.9 Mastermind

Mastermind merupakan permainan papan yang dimainkan oleh dua orang mengenai pemecahan kode yang diciptakan oleh Mordecai Meirowitz, warga Israel yang bekerja sebagai tukang pos dan ahli telekomunikasi [10]. Pada permainan ini pertama kedua pemain menentukan ronde yang akan dimainkan namun ronde yang dimainkan haruslah genap. Setelah itu di tiap ronde, seseorang akan menjadi *codemaster* dan *codebreaker*. *Codemaster* bertugas sebagai orang yang membuat kode, sedangkan *codebreaker* harus menebak kode yang telah ditentukan oleh *codemaster*. Tiap pemain mendapat kesempatan untuk menjadi *codemaster* dan *codebreaker* yang sama. Kode yang dibuat berupa empat urutan warna. Ketika *codebreaker* menebak, maka *codemaster* akan memberi tanda mengenai jawaban *codebreaker*. Warna putih melambangkan warna pada jawaban ada dalam kode dan warna putih melambangkan posisi warna dari tebakan *codebreaker* sudah benar. *Codebreaker* memiliki dua belas kali kesempatan menjawab. Jika kode berhasil dijawab maka *codebreaker* memenangkan ronde dan mendapat satu poin, dan jika kode tidak berhasil dijawab maka *codemaster* memenangkan ronde dan mendapatkan poin.

*(Halaman ini sengaja dikosongkan)*

## **BAB III**

### **PERANCANGAN SISTEM**

Pada tugas akhir ini, akan difokuskan pada fitur mini game yang ada pada game “AVIAR”. Pada bab ini akan menjelaskan tentang analisis dan perancangan mini game yang ada pada game “AVIAR” dengan menggunakan Oculus Rift. Pembahasan yang dilakukan meliputi analisis kebutuhan fungsionalitas dan perancangan perangkat lunak.

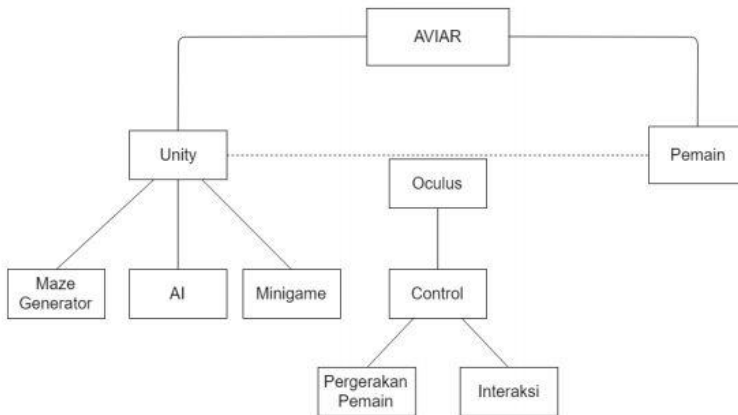
#### **3.1 Deskripsi Umum Perangkat Lunak**

Pada tugas akhir ini dibuat sebuah *game* puzzle berupa labirin yang bisa berubah – ubah pola dalam jangka waktu yang telah ditentukan. Aplikasi ini awalnya terinspirasi dari film Maze Runner. Maze Runner sendiri merupakan sebuah film yang diangkat dari sebuah novel karya James Dashner pada tahun 2009. Film ini menceritakan sebuah dunia dimana ada orang – orang yang dipilih untuk dijadikan semacam tikus percobaan. Mereka ditempatkan di dalam sebuah labirin yang dapat berubah-ubah pola. Selain itu pada malam hari, di dalam labirin akan dikeluarkan sebuah makhluk yang difungsikan untuk membunuh orang – orang dalam labirin yang mencoba untuk keluar. Ketika menonton film tersebut, saya berfikir apa yang sebenarnya dirasakan oleh orang – orang tersebut di dalam labirin.

Maka dari itu, penulis berencana untuk membuat sebuah *game* berbasis VR dengan tujuan agar bisa memberikan pengalaman yang lebih nyata dan imersif bagi pemain. Pemain akan ditempatkan di dalam sebuah labirin dan memiliki tujuan untuk dapat keluar dari labirin tersebut. Posisi matahari dalam *game* ini dapat berubah layaknya di dunia nyata sehingga ada siang dan malam. Ketika setiap kali hari berubah menjadi siang, maka labirin akan berubah pola secara acak. Dan ketika hari berubah menjadi malam, maka musuh akan di tempatkan di dalam labirin untuk menghadang pemain dalam melarikan diri. Untuk bisa

melawan musuh yang ada, pemain bisa mengambil barang yang dapat membantu pemain untuk keluar dari labirin. Barang tersebut bisa didapatkan dari dalam peti yang tersebar di dalam labirin. Barang yang disediakan bisa berupa senjata atau peningkatan status dari pemain. Status yang dimaksudkan adalah kecepatan bergerak, ketahanan terhadap serangan musuh, jumlah *health*, dan lain – lain. Untuk bisa membuka peti tersebut, pemain harus menyelesaikan mini game yang terletak berdampingan dengan masing – masing peti.

### 3.2 Arsitektur Permainan



**Gambar 3.1 Arsitektur Game AVIAR**

Gambar 3.1 merupakan arsitektur *game* AVIAR secara keseluruhan. AVIAR merupakan salah satu bentuk interaksi antara manusia dan komputer. Manusia disini digambarkan oleh pemain sedangkan komputer digambarkan oleh Unity. Unity bertugas untuk menjalankan permainan AVIAR yang terdiri dari tiga komponen utama yaitu *Maze Generator*, *AI (Artificial Intelligence)*, dan *Minigame*. Lalu untuk menjembatani antara *game* dan pemain digunakan Oculus. Oculus sendiri digunakan pemain untuk bisa bergerak dan melihat di dalam dunia *Virtual*

*Reality*. Pemain akan menggunakan kontroler dari Oculus untuk bisa bergerak dan berinteraksi dengan *minigame*, benda, dan menu di dalam *game*. Interaksi yang bisa dilakukan adalah memilih dan menggenggam benda.

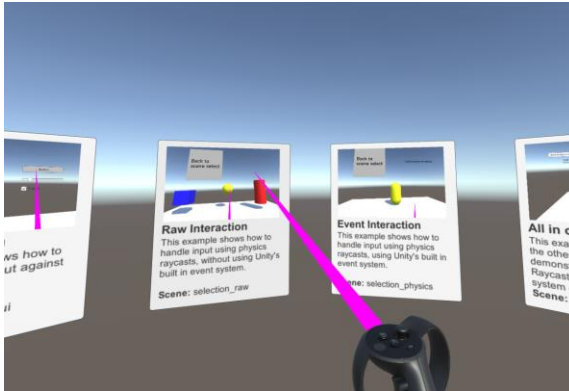
Pada tugas akhir ini akan difokuskan pada pembuatan *minigame* dan interaksi pada kontroler.

### **3.3 Perancangan Skenario Permainan**

Sub bab ini membahas rancangan alur bermain pada mini game yang terdapat di dalam maze dan apa saja yang dibutuhkan agar *minigame* dapat dimainkan. Rancangan skenario yang dibahas meliputi tiga jenis mini game yaitu Legot, Picture Memory, dan Tap Fast. Untuk masing – masing mini game akan dijelaskan aturan main, cara bermain, tingkat kesulitan dan kondisi menang atau kalah. Disamping itu juga akan dijelaskan mengenai kontroler sebagai kebutuhan untuk memainkan *minigame*.

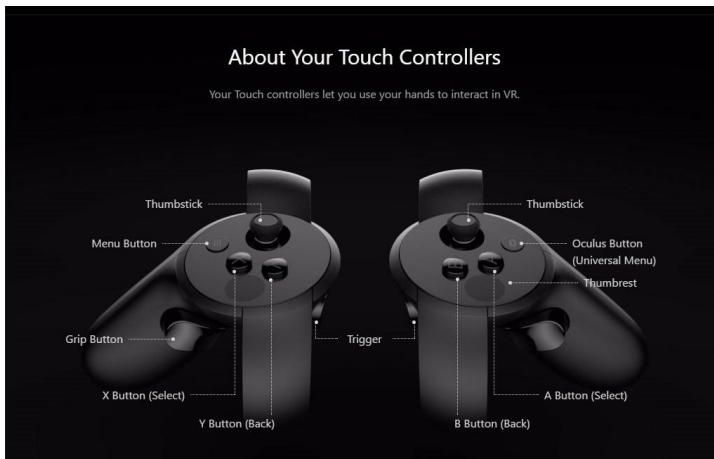
#### **3.3.1 Kontroler**

Dalam memainkan *minigame* pada game AVIAR, diperlukan kontroler berupa oculus touch. Kontroler ini akan berperan penting dalam memilih atau menekan objek – objek pada *minigame*. Cara memilih objek adalah dengan mengarahkan laser yang ada di kontroler kanan seperti pada Gambar 3.2. Lalu ketika laser telah mengarah ke benda yang dimaksud, tekan tombol trigger. Tombol – tombol pada oculus touch dapat dilihat pada Gambar 3.3



**Gambar 3.2 Laser pada oculus touch**

(Sumber: <https://developer.oculus.com/blog/easy-controller-selection/>)



**Gambar 3.3 Tombol-tombol pada oculus touch**

(Sumber: <https://www.roadtovr.com/oculus-touch-review-reach-rift/>)

Dari penjelasan pada paragraf sebelumnya bisa diambil kebutuhan fungsionalitas dari kontroler yang dijelaskan pada Tabel 3.1.

**Tabel 3.1 Kebutuhan fungsionalitas Kontroler**

<b>Kode</b>	<b>Kebutuhan Fungsionalitas</b>
F-001	Kontroler dapat mendeteksi objek yang dipilih menggunakan laser

### 3.3.2 Legot

Legot merupakan permainan puzzle yang melatih logika dari pemain. Dalam mini game ini pemain harus menebak urutan angka dari pertanyaan yang tertera. Pertanyaan yang tertera akan diambil secara acak dari file xml yang telah disiapkan. Di dalam file xml pertanyaan telah dibagi menjadi tiga kategori yaitu mudah, sedang, dan sulit. *Minigame* akan mengambil secara acak dari kategori tersebut sesuai dengan tingkat kesulitan yang telah ditentukan. Angka yang harus ditebak bisa berupa tiga atau empat digit angka tergantung pada kesulitannya.

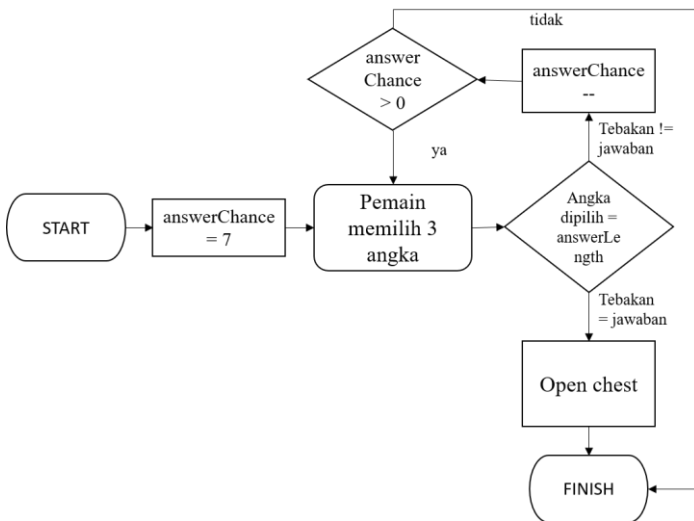
Aturan dan cara bermain pada *minigame* ini adalah pertama, pertanyaan yang tersedia berupa urutan – urutan angka beserta status kebenaran dari urutan angka tersebut. Yang dimaksud dengan status kebenaran angka adalah jumlah angka yang benar dan posisinya. Angka yang benar dilambangkan dengan huruf “A” sedangkan posisi yang benar dilambangkan dengan huruf “P”. Pemain diberikan kesempatan menjawab sejumlah tujuh kali. Untuk menebak urutan angka, pemain diberikan tombol yang bertuliskan angka nol hingga sembilan. Pemain harus memasukkan kombinasi angka yang benar pada tombol–tombol angka dengan menekan tombol–tombol tersebut. Setelah melakukan percobaan dalam menebak, akan muncul hasil dari tebakan pemain. Dalam hasil tersebut juga akan tertera status kebenaran dari jawaban pemain. Pada *minigame* ini terdapat tiga tingkat kesulitan yaitu mudah, sedang, dan sulit. Faktor yang

mempengaruhi tingkat kesulitan pada *minigame* ini adalah dari jumlah angka kombinasi dan petunjuk yang diberikan.

Pada *minigame* ini terdapat tiga kesulitan, yaitu mudah, sedang, dan sulit. Tingkat kesulitan pada minigame ini dipengaruhi oleh dua faktor, yaitu petunjuk dan jumlah kombinasi angka. Pada tingkat kesulitan mudah, petunjuk yang diberikan mudah dipahami dan jumlah kombinasi angka hanya tiga angka. Untuk tingkat kesulitan sedang, merupakan kombinasi dari kedua faktor tersebut. Ada yang memiliki petunjuk yang mudah dimengerti namun terdiri dari empat kombinasi angka dan petunjuk yang lebih sulit namun terdiri dari tiga angka. Lalu tingkat kesulitan sulit memiliki petunjuk yang lebih sulit dimengerti dan terdiri dari empat kombinasi angka.

Kondisi menang tercapai ketika pemain dapat menebak urutan angka yang benar. Kondisi kalah akan terjadi apabila kesempatan menjawab sudah habis.

Alur dari permainan ini dapat dilihat pada Gambar 3.4 dibawah.



**Gambar 3.4 Flowchart *minigame* Legot**



Dari aturan dan cara bermain yang sudah ada dapat diambil kebutuhan fungsionalitas seperti pada Tabel 3.2 dibawah.

**Tabel 3.2 Kebutuhan fungsionalitas *minigame* Legot**

<b>Kode</b>	<b>Kebutuhan Fungsionalitas</b>
F-002	<i>Minigame</i> dapat mengambil soal pada text file yang tersedia
F-003	<i>Minigame</i> Legot dapat memasukkan angka pada tombol yang ditekan sesuai dengan urutan pada kombinasi angka.
F-004	<i>Minigame</i> Legot dapat menghitung status kebenaran dari player.
F-005	<i>Minigame</i> Legot dapat membuka peti ketika pemain menjawab dengan benar.

### 3.3.3 Picture Memory

Picture Memory merupakan permainan yang mengasah kemampuan memori pemain. Dalam *minigame* ini, pemain harus mencari pasangan dari gambar – gambar yang ada.

Aturan dan cara bermain pada *minigame* ini adalah pemain diberikan panel – panel yang memiliki gambar. Namun panel tersebut berada dalam posisi terbalik sehingga gambar membelakangi pemain. Untuk membalik dan mencari pasangannya, pemain memilih dua panel yang ingin diputar dan dicocokkan. Jika panel yang dipilih memiliki gambar yang berbeda, maka panel akan berputar kembali ke posisi membelakangi pemain. Namun jika memiliki gambar yang sama, panel akan diam dalam posisi gambar menghadap pemain yang menandakan bahwa pemain berhasil mencocokkan gambar dan mendapat satu poin. Pada *minigame* ini terdapat tiga tingkat kesulitan yaitu mudah, sedang, dan sulit. Faktor yang mempengaruhi tingkat kesulitan pada *minigame* ini adalah dari jumlah gambar yang harus dicocokkan.

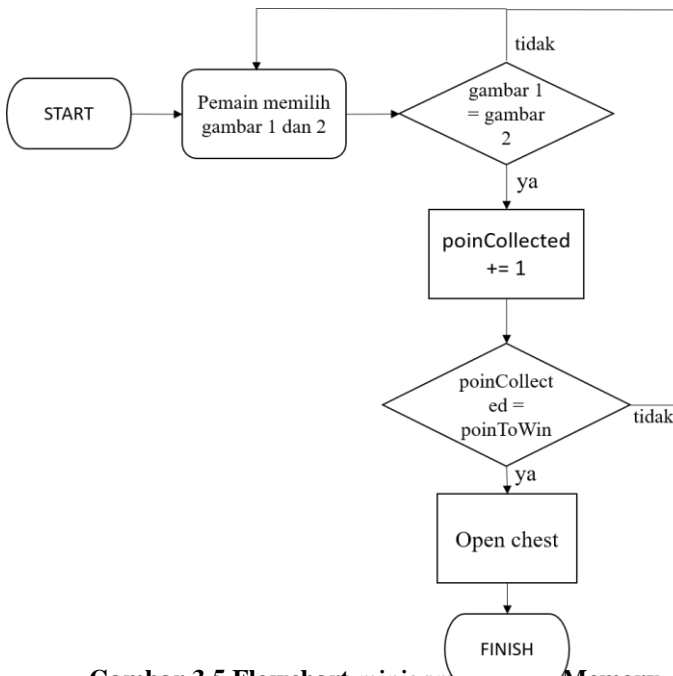
Pada *minigame* ini terdapat tiga kesulitan, yaitu mudah, sedang, dan sulit. Tingkat kesulitan dibedakan berdasarkan

jumlah panel. Pada tingkat kesulitan mudah delapan panel, sedang dua belas panel, dan sulit enam belas panel.

Kondisi menang tercapai apabila semua gambar telah dicocokkan. Dengan tercapainya kondisi menang maka peti yang berada pada *minigame* akan terbuka. Pada *minigame* ini tidak terdapat kondisi kalah.

Untuk men-*generate* minigame ini, objek Picture Memory dipasang panel sejumlah nilai maksimumnya yaitu enam belas. Ketika men-*generate* tingkat kesulitan mudah dan sedang, maka panel akan di non-aktifkan hingga panel yang tersisa berjumlah panel yang dibutuhkan. Namun panel yang di non-aktifkan sudah ditentukan agar semua panel memiliki pasangannya masing-masing.

Alur dari permainan ini dapat dilihat pada Gambar 3.5 dibawah.



**Gambar 3.5 Flowchart *minigame Picture Memory***

Dari deskripsi aturan dan cara bermain yang sudah ada dapat diambil kebutuhan fungsionalitas seperti pada Tabel 3.3 dibawah.

**Tabel 3.3 Kebutuhan fungsionalitas *minigame* Picture Memory**

<b>Kode</b>	<b>Kebutuhan Fungsionalitas</b>
F-006	<i>Minigame</i> dapat memutar panel ketika ditekan.
F-007	<i>Minigame</i> dapat mencocokkan gambar pada panel pertama dan kedua yang ditekan.
F-008	<i>Minigame</i> dapat membuka peti ketika semua panel sudah dalam keadaan terbalik.

### 3.3.4 Tap Fast

Tap Fast merupakan permainan yang melatih reflek dari pemain. Dalam mini game ini pemain harus memencet silinder yang keluar dari tembok.

Aturan dan cara bermain pada *minigame* adalah pemain pertama harus menekan tombol start yang terletak pada bagian bawah *minigame*. Dengan melakukan hal tersebut, pemain telah menginisiasi sebuah game yang terdiri dari beberapa ronde. Selama ronde sedang berjalan, silinder – silinder akan bergerak keluar dari tembok. Ketika silinder sedang berada diluar tembok, pemain harus menekan silinder tersebut sebelum kembali masuk. Dengan menekan silinder, pemain mendapatkan satu poin. Pada *minigame* ini terdapat tiga tingkat kesulitan yaitu mudah, sedang, dan sulit. Pada *minigame* ini terdapat tiga tingkat kesulitan yaitu mudah, sedang dan sulit. Faktor yang mempengaruhi tingkat kesulitan pada *minigame* ini adalah dari kecepatan gerakan silinder saat keluar dan masuk dan ukuran silinder.

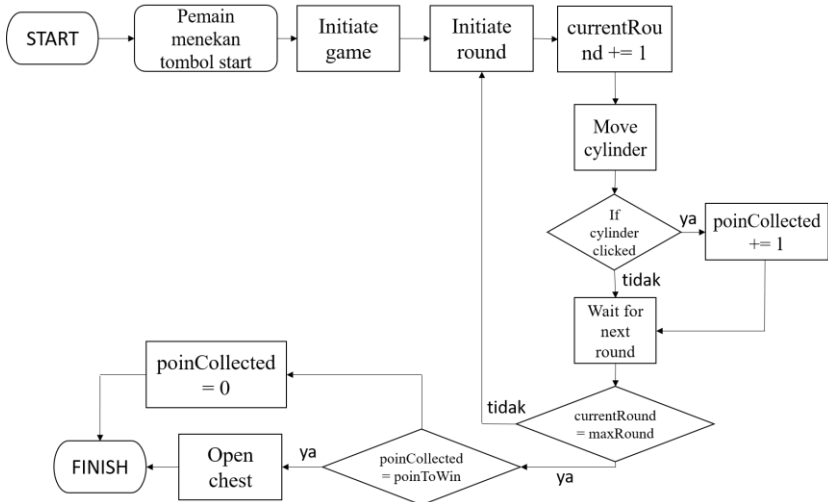
Pada *minigame* ini terdapat tiga kesulitan, yaitu mudah, sedang, dan sulit. Tingkat kesulitan dibedakan berdasarkan jumlah minimal silinder yang harus ditekan, jumlah silinder yang keluar, jumlah ronde, dan kecepatan gerak silinder. Tingkat kesulitan Tap Fast akan dijelaskan pada Tabel 3.4 dibawah.

**Tabel 3.4 Pembagian tingkat kesulitan *minigame* Tap Fast**

Tingkat Kesulitan	Mudah	Sedang	Sulit
Silinder yang harus ditekan (buah)	6	16	20
Jumlah silinder (buah)	7	12	17
Jumlah silinder yang keluar per ronde (buah)	1 - 3	2 - 4	2 - 6
Jumlah Ronde	6	8	10
Ukuran silinder (unit pada satuan unity)	1	0,85	0,7

Kondisi menang akan tercapai ketika pemain mendapatkan poin yang lebih atau sama dengan ketentuan dari kesulitan *minigame*. Kondisi kalah akan terjadi jika pemain mendapat poin dibawah ketentuan dari kesulitan *minigame*. Namun walaupun pemain gagal, pemain dapat bermain kembali dengan menekan tombol start.

Alur dari permainan ini dapat dilihat pada Gambar 3.6 dibawah.



**Gambar 3.6 Flowchart *minigame* Tap Fast**

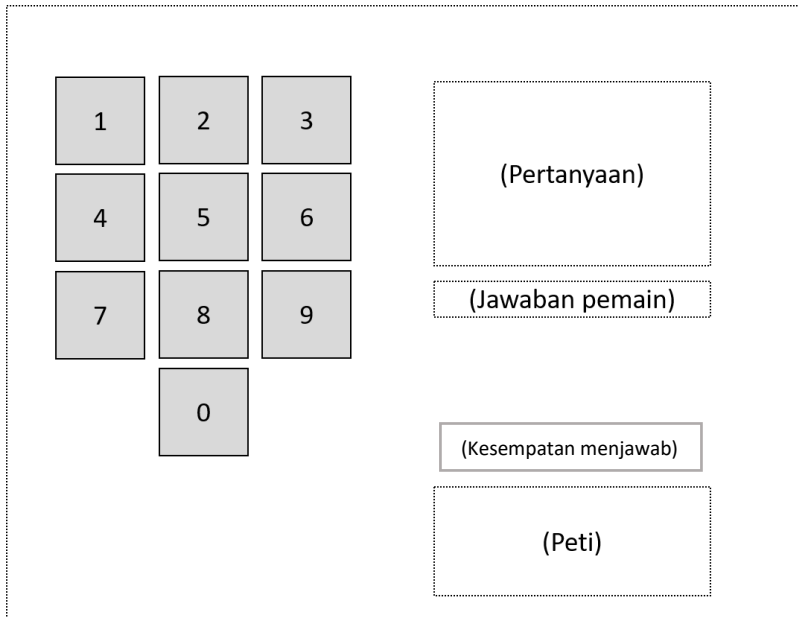
Dari deskripsi aturan dan cara bermain yang sudah ada dapat diambil kebutuhan fungsionalitas seperti pada Tabel 3.4 dibawah.

**Tabel 3.5 Kebutuhan fungsionalitas *minigame* Tap Fast**

Kode	Kebuthan Fungsionalitas
F-009	<i>Minigame</i> dapat memulai permainan ketika tombol start ditekan.
F-010	<i>Minigame</i> dapat menggerakkan silinder.
F-011	<i>Minigame</i> dapat menghitung jumlah silinder yang ditekan.
F-012	<i>Minigame</i> dapat membuka peti ketika jumlah silinder yang ditekan telah mencapai ketentuan

### 3.4 Perancangan Antarmuka

#### 3.4.1 Legot



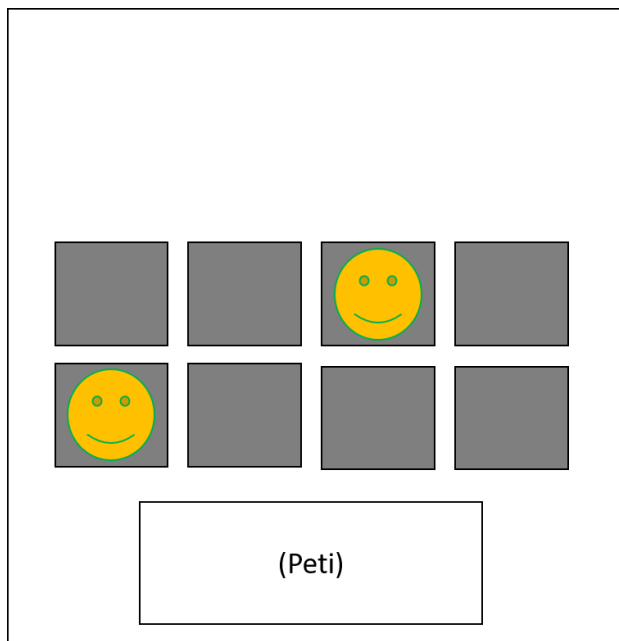
**Gambar 3.7 Antarmuka *minigame* Legot**

Gambar 3.6 menggambarkan antarmuka dari *minigame* legot. Seperti pada Gambar 3.6 bisa dilihat bahwa

1. Di sebelah kiri terdapat kotak dengan angka di dalamnya. Kotak tersebut merupakan tombol yang akan digunakan pemain untuk memasukkan jawaban.
2. Di sebelah dari kumpulan tombol angka, terdapat pertanyaan yang menjadi petunjuk untuk pemain dalam menebak jawaban yang benar.
3. Di bawah dari pertanyaan akan dimunculkan jawaban dari pemain ketika digit yang dimasukan sudah sesuai.

4. Di bawah jawaban akan dimunculkan sisa kesempatan pemain untuk menjawab kombinasi angka.
5. Peti yang berada di bawah adalah peti yang menyimpan bantuan untuk pemain sebagai hadiah jika berhasil menyelesaikan *minigame*.

### 3.4.2 Picture Memory



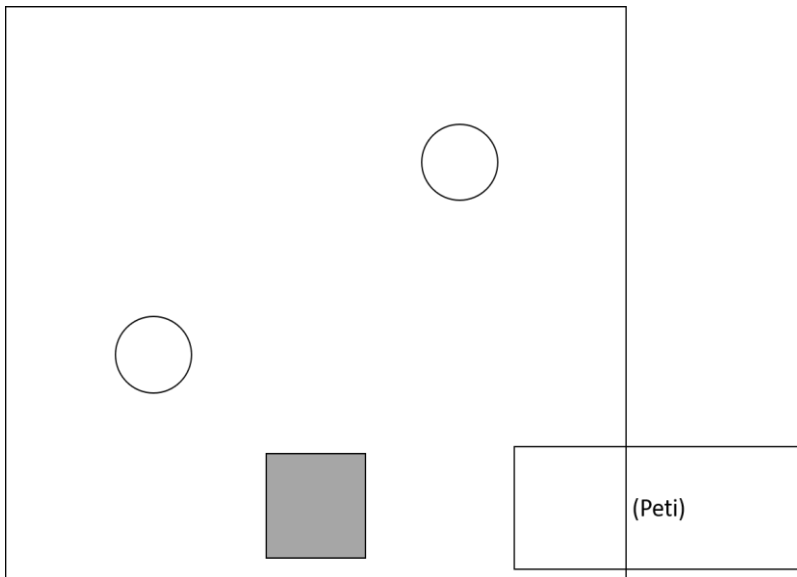
**Gambar 3.8 Antarmuka *minigame* Picture Memory**

Gambar 3.7 menggambarkan antarmuka dari *minigame* picture memory. Seperti pada Gambar 3.7 bisa dilihat bahwa

1. Kotak berwarna abu – abu menggambarkan posisi awal panel.

2. Kotak berwarna abu – abu dengan gambar *smiley* merupakan tampilan panel dalam posisi sudah dibalik. Hal ini terjadi ketika pemain memilih panel tersebut.
3. Peti yang berada di bawah adalah peti yang menyimpan bantuan untuk pemain sebagai hadiah jika berhasil menyelesaikan *minigame*.

### 3.4.3 Tap Fast



**Gambar 3.9 Antarmuka *minigame* Tap Fast**

Gambar 3.8 menggambarkan antarmuka dari *minigame* tap fast. Seperti pada Gambar 3.8 bisa dilihat bahwa

1. Kotak berwarna abu – abu terletak di bawah merupakan tombol untuk memulai permainan.
2. Dua lingkaran menggambarkan silinder yang keluar dari tembok.



3. Peti yang berada di bawah kanan adalah peti yang menyimpan bantuan untuk pemain sebagai hadiah karena jika menyelesaikan *minigame*.

*(Halaman ini sengaja dikosongkan)*

## **BAB IV IMPLEMENTASI**

Pada bab ini akan dibahas mengenai implementasi dari perancangan aplikasi. Di dalamnya mencakup proses penerapan dan pengimplementasian dalam bentuk *code*, dan antarmuka yang mengacu pada rancangan yang telah dibahas sebelumnya.

### **4.1 Lingkungan Implementasi**

Lingkungan implementasi dari tugas akhir ini adalah perangkat desktop dengan spesifikasi sebagai berikut:

**Tabel 4.1 Spesifikasi lingkungan implementasi**

<b>Perangkat</b>	<b>Spesifikasi</b>
Perangkat Keras	<ul style="list-style-type: none"><li>• Prosesor Intel(R) Core (TM) i7-770 CPU @ 3.60GHz (8 CPUs), ~3.6 GHz</li><li>• RAM 8 Gigabyte</li><li>• Graphic card NVIDIA GeForce GTX 1060 3GB</li><li>• Oculus Rift</li></ul>
Perangkat Lunak	<ul style="list-style-type: none"><li>• Sistem operasi Windows 10 Home Single Language 64-bit</li><li>• Unity 2017.3.0f3 Personal (64-bit)</li></ul>

### **4.2 Implementasi pada Permainan**

Implementasi dari masing – masing *minigame* dituliskan menggunakan bahasa C#. Pada sub bab ini akan membahas objek pada *minigame* dan komunikasinya antar kelas.

#### 4.2.1 Penempatan *Minigame* Pada Labirin

Terdapat dua metode penempatan *minigame* yang dilakukan, yaitu secara manual dan otomatis. Untuk menentukan jumlah *minigame* yang akan diletakkan di dalam labirin, penulis melakukan perhitungan secara manual dengan melihat ukuran labirin di tiap levelnya. Selain itu penulis juga menentukan jumlah kesulitan dan jumlah per jenis *minigame* yang akan ditempatkan. Setelah itu *minigame* akan ditempatkan secara otomatis dengan membagi labirin ke beberapa segmen. Pembagian segmen dibagi menjadi dua tipe, yaitu membagi per 2x2 untuk ukuran labirin lebih kecil dari 6x6 dan 3x3 untuk labirin lebih besar dari 6x6. Kemudian per segmen akan ditempatkan satu *minigame*. Metode penempatan secara manual dilakukan dengan menempatkan di dalam scene unity secara langsung. Metode ini digunakan untuk level 1 hingga 15. Hal ini dilakukan karena pola dari pagi dan malam sudah ditentukan, maka dengan menempatkan manual maka *minigame* ditempatkan yang tidak menghalangi jalan pemain. Untuk metode penempatan secara otomatis diimplementasi untuk level *endless*. Hal ini dilakukan untuk menghindari *minigame* ditempatkan ditempat yang sama atau terlalu berdekatan. Penempatan *minigame* secara otomatis dilakukan dengan menggunakan kelas *MinigameSpawner*. Ketika scene level *diload* maka kelas ini akan menempatkan *minigame* sesuai dengan tempat yang telah diberikan. Kelas *MinigameSpawner* dapat dilihat pada Kode Sumber 4.1

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class MinigameSpawner : MonoBehaviour {
6.
7.     private Vector3[] minigamePos;
8.     private Vector3[] minigameRot;
9.     private MazeSpawner msComponent;
10.    private int columns;
11.    private int rows;

```

```

12.     private float minigameOffset;
13.     private int numberOfMG;
14.
15.     public GameObject mazeSpawner;
16.     public GameObject[] minigames = new GameObject
t[5];
17.     public int[] numberOfDif = new int[3];    //0
        = easy, 1 = med, 2 = hard
18.     public int[] numberOfMGType = new int[3]; //0
        = legot, 1 = pm, 2 = tf
19.     //jumlah numberOfDif dan numberOfTypeMG HARUS
        SAMA
20.
21.     void Awake () {
22.         numberOfMG = 0;
23.
24.         for (int a = 0; a<3; a++)
25.         {
26.             numberOfMG = numberOfMG + numberOfMGT
ype[a];
27.         }
28.
29.         //Assign maze size from maze spawner
30.         msComponent = mazeSpawner.GetComponent<Ma
zeSpawner>();
31.         columns = msComponent.Columns;
32.         rows = msComponent.Rows;
33.
34.         //Assign rotation and position
35.         minigameRot = new Vector3(numberOfMG);
36.         minigamePos = new Vector3(numberOfMG);
37.         assignPosAndRot(numberOfMG);
38.
39.         //Instantiate Minigame
40.         int mgNumber = 0;
41.         for(int a = 0; a<3; a++)
42.         {
43.             for(int b = 0; b<numberOfMGType[a]; b
        ++
44.             {
45.                 instantiateMinigame(a,mgNumber);

```

```

46.         mgNumber = mgNumber + 1;
47.     }
48. }
49. }
50.
51. int assignDifficulty()
52. {
53.     int Adifficulty;
54.
55.     Adifficulty = randomDifficulty();
56.     minusDifficulty(Adifficulty);
57.     return Adifficulty;
58. }
59.
60. int randomDifficulty()
61. {
62.     int difficulty;
63.     difficulty = Random.Range(2, 5); //2 = ea
sy, 3 = med, 4 = hard
64.
65.     switch (difficulty)
66.     {
67.         case 2:
68.             if(numberOfDif[0] == 0)
69.             {
70.                 difficulty = randomDifficulty
71.             (
72.                 );
73.             }
74.             break;
75.         case 3:
76.             if (numberOfDif[1] == 0)
77.             {
78.                 difficulty = randomDifficulty
79.             (
80.                 );
81.             }
82.             break;
83.         case 4:
84.             if (numberOfDif[2] == 0)
85.             {
86.                 difficulty = randomDifficulty
87.             (
88.                 );
89.             }
90.             break;
91.     }
92. }

```

```

84.             break;
85.         }
86.
87.         return difficulty;
88.     }
89.
90.     void minusDifficulty(int difficulty)
91.     {
92.         switch (difficulty)
93.         {
94.             case 2:
95.                 numberOfDif[0] = numberOfDif[0] -
96.                 1;
97.                 break;
98.             case 3:
99.                 numberOfDif[1] = numberOfDif[1] -
100.                1;
101.                break;
102.            case 4:
103.                numberOfDif[2] = numberOfD
104.                if[2] - 1;
105.                break;
106.            }
107.        }
108.
109.        void instantiateMinigame(int mgType, i
110.        nt mgNumber)
111.        {
112.            int difficulty;
113.            switch (mgType)
114.            {
115.                case 0:
116.                    GameObject legot;
117.                    LegotGameController legotC
118.                    omponent;
119.                    difficulty = assignDifficu
120.                    lty();
121.                    legot = Instantiate(miniga
122.                    mes[0], minigamePos[mgNumber], Quaternion.Euler(m
123.                    inigameRot[mgNumber]));
124.                    legotComponent = legot.Get
125.                    Component<LegotGameController>();

```

```

117.         legotComponent.difficulty
        = difficulty;
118.         break;
119.     case 1:
120.         GameObject pm;
121.         PictureMemoryMaster pmComp
        onent;
122.         difficulty = assignDifficu
        lty();
123.         pm = Instantiate(minigames
        [1], minigamePos[mgNumber], Quaternion.Euler(mini
        gameRot[mgNumber]));
124.         pmComponent = pm.GetCompon
        ent<PictureMemoryMaster>();
125.         pmComponent.difficulty = d
        ifficulty;
126.         break;
127.     case 2:
128.         GameObject tf;
129.         difficulty = assignDifficu
        lty();
130.         tf = Instantiate(minigames
        [difficulty], minigamePos[mgNumber], Quaternion.E
        uler(minigameRot[mgNumber]));
131.         break;
132.     }
133. }
134.
135. void assignPosAndRot(int mgToAssign)
136. {
137.     for(int a = 0; a<mgToAssign; a++)
138.     {
139.         //Assign rotation
140.         int randomRotation;
141.         randomRotation = Random.Range(
        0, 4) * 90;
142.         minigameRot[a] = new Vector3(0
        , randomRotation, 0);
143.
144.         //Assign position and offset a
        ccording to rotation

```



```

145.         if (randomRotation == 180 || r
    andomRotation == 270)
146.         {
147.             minigameOffset = -1.7f;
148.         }
149.         else if (randomRotation == 0 |
    | randomRotation == 90)
150.         {
151.             minigameOffset = 1.7f;
152.         }
153.         assignPosition(randomRotation,
    a,minigameOffset);
154.         if (rows >= 6 || columns >= 6)
            break;
155.     }
156. }
157.
158. void assignPosition(int rotation, int
    mgNumber, float offset)
159. {
160.     int randomX;
161.     int randomZ;
162.
163.     if (columns == 4 && rows == 4)
164.     {
165.         //khusus 4x4
166.         if(mgNumber == 0)
167.         {
168.             randomX = Random.Range(0,
    2) * 4;
169.             randomZ = Random.Range(2,
    rows) * 4;
170.             assignPosByRot(randomX, ra
    ndomZ, mgNumber, rotation, offset);
171.         }
172.         else if(mgNumber == 1)
173.         {
174.             randomX = Random.Range(2,
    columns) * 4;
175.             randomZ = Random.Range(0,
    2) * 4;

```

```

176.             assignPosByRot(randomX, ra
ndomZ, mgNumber, rotation, offset);
177.         }
178.     }
179.     else if (columns < 6 || rows < 6)
180.     {
181.         //untuk 4x5 sampe 5x6
182.         if(mgNumber == 0)
183.         {
184.             randomX = Random.Range(0,
2) * 4;
185.             randomZ = Random.Range(2,
rows) * 4;
186.             assignPosByRot(randomX, ra
ndomZ, mgNumber, rotation, offset);
187.         }
188.         else if(mgNumber == 1)
189.         {
190.             randomX = Random.Range(2,
columns) * 4;
191.             randomZ = Random.Range(0,
2) * 4;
192.             assignPosByRot(randomX, ra
ndomZ, mgNumber, rotation, offset);
193.         }
194.         else if(mgNumber == 2)
195.         {
196.             randomX = Random.Range(2,
columns) * 4;
197.             randomZ = Random.Range(2,
rows - 1) * 4;
198.             assignPosByRot(randomX, ra
ndomZ, mgNumber, rotation, offset);
199.         }
200.     }
201.     else
202.     {
203.         //untuk 6x6 keatas
204.         int counter = 0;
205.         for(int a = 0; a<(rows/3); a++)
    )

```

```

206.         {
207.             for(int b = 0; b<(columns/
208.                 3); b++)
209.             {
210.                 if(a == 0)
211.                 {
212.                     b++;
213.                 }
214.                 int randomRotation;
215.                 randomRotation = Rando
216.                     m.Range(0, 4) * 90;
217.                 minigameRot[counter] =
218.                     new Vector3(0, randomRotation, 0);
219.                 if (randomRotation ==
220.                     180 || randomRotation == 270)
221.                 {
222.                     minigameOffset = -
223.                         1.7f;
224.                 }
225.                 else if (randomRotatio
226.                     n == 0 || randomRotation == 90)
227.                 {
228.                     minigameOffset = 1
229.                         .7f;
230.                 }
231.                 randomX = Random.Range
232.                     (a * 3, (a * 3) + 3) * 4;
233.                 randomZ = Random.Range
234.                     (b * 3, (b * 3) + 3) * 4;
235.                 assignPosByRot(randomX
236.                     , randomZ, counter, randomRotation, minigameOffse
237.                     t);
238.                 counter++;
239.             }
240.         }
241.     }
242. }
243.
244. void assignPosByRot(int x, int z, int
245.     mgNumber, int rotation, float offset)

```

```

236.         {
237.             if (rotation == 0 || rotation == 1
238.                 80)
239.             {
240.                 minigamePos[mgNumber] = new Ve
241.                 ctor3(x, 1, (z + offset));
242.             }
243.             else if (rotation == 90 || rotatio
244.                 n == 270)
245.             {
246.                 minigamePos[mgNumber] = new Ve
247.                 ctor3((x + offset), 1, z);
248.             }
249.         }

```

**Kode Sumber 4.1 Kelas MinigameSpawner**

#### 4.2.2 Kontroler

Objek Kontroler adalah objek yang memungkinkan pemain untuk berinteraksi dengan tombol-tombol pada *minigame* menggunakan laser yang ada pada kontroler. Objek ini memiliki kelas *MinigameController*. Kelas ini mengambil tag dari game objek yang ditekan. Jika tag dari game objek tersebut sesuai, maka kelas ini akan memberikan sinyal kepada tombol yang ditekan agar teraktifasi. Fungsi pada kelas ini akan dipanggil oleh Kelas *OVRRawRaycast* yang merupakan kelas bawaan dari Oculus SDK. Kelas *MinigameInteraction* dapat dilihat pada Kode Sumber 4.2.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class MinigameInteraction : MonoBehaviour
6. {

```

```

7.     public void OnHoverEnter(Transform t)
8.     {
9.
10.    }
11.
12.    public void OnHoverExit(Transform t)
13.    {
14.
15.    }
16.
17.    public void OnSelected(Transform t)
18.    {
19.        GameObject gjHovered;
20.        gjHovered = t.gameObject;
21.
22.        if(t.gameObject.tag == "PMBButton") //PictureMemory
23.        {
24.            PictureMemoryController PMButtonComponent;
25.
26.            PMButtonComponent = gjHovered.GetComponent<PictureMemoryController>();
27.            PMButtonComponent.isActive = true;
28.
29.        }
30.        else if(t.gameObject.tag == "LegotButtonChild") //Legot
31.        {
32.            LegotInput LegotButtonComponent;
33.
34.            LegotButtonComponent = gjHovered.GetComponent<LegotInput>();
35.            LegotButtonComponent.isActive = true;
36.        }
37.        else if(t.gameObject.tag == "TFStartButton")
38.        {
39.            TapFastStart TFStartComponent;
40.

```

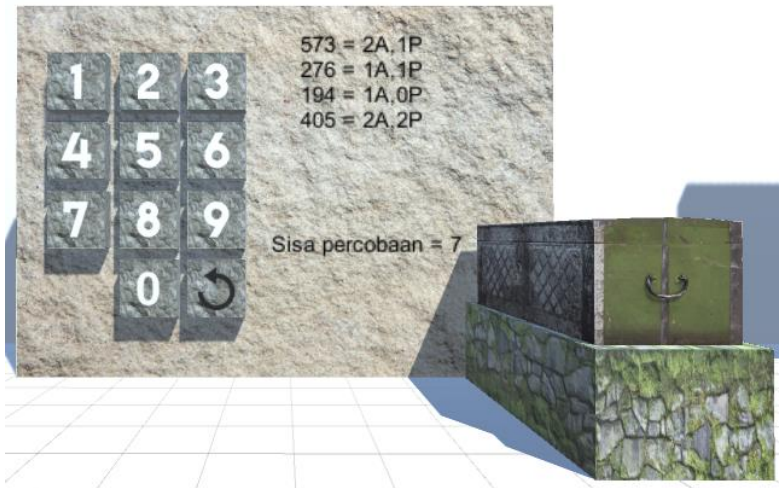
```

41.         TFStartComponent = gjHovered.GetCompo
nent<TapFastStart>();
42.         TFStartComponent.isActive = true;
43.     }
44.     else if(t.gameObject.tag == "TFButton")
45.     {
46.         TapFastButton TFButtonComponent;
47.
48.         TFButtonComponent = gjHovered.GetComp
onent<TapFastButton>();
49.         TFButtonComponent.isClicked = true;
50.     }
51.     else
52.     {
53.         Debug.Log("Object selected doesn't ha
ve the required tag");
54.     }
55.
56.
57.         Debug.Log(t.gameObject.name + " clicked")
;
58.     }
59.
60. }

```

**Kode Sumber 4.2 Kelas MinigameInteraction**

### 4.2.3 Legot



**Gambar 4.1 Implementasi *minigame* Legot**

Gambar 4.1 merupakan hasil implementasi dari *minigame* legot. Pada *minigame* ini terdapat lima script pada masing – masing komponen yang saling berkomunikasi. Komponen tersebut adalah *minigame* secara keseluruhan yaitu objek Legot, Legot Button, Soal Legot, Jawaban Player, dan Data Controller.

#### 4.2.3.1 Legot

Objek legot merupakan kumpulan dari komponen – komponen pembentuknya. Objek ini memiliki kelas LegotGameController yang berfungsi sebagai penjemputan untuk objek – objek lainnya. LegotGameController akan menerima data angka yang dikirimkan oleh tombol angka ketika ditekan lalu menyimpannya dalam sebuah array of string. Dari array tersebut akan diubah ke dalam string yang nantinya akan ditampilkan oleh objek Jawaban Player. Selain itu, kelas

ini juga menyimpan jumlah benar dari angka dan posisi dari jawaban pemain. Kelas ini juga menyimpan jumlah kombinasi angka bergantung pada soal yang diload pada *minigame* tersebut. Ketika pemain menjawab dengan benar, maka kelas ini akan memberikan sinyal kepada objek Peti untuk membuka. Fungsi yang terakhir adalah menghitung sisa kesempatan pemain untuk menjawab, apabila kesempatan habis maka pemain tidak bisa memasukan jawaban lagi. Kelas LegotGameController dapat dilihat pada Kode Sumber 4.3 di bawah.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using System;
5.
6. public class LegotGameController : MonoBehaviour
7. {
8.     private LoadSoal dataLoadSoal;
9.     private ChestController legotChestController;
10.
11.     private PrintJawabanPlayer printComponent;
12.
13.     public GameObject chestLegot;
14.     public GameObject soalLegot;
15.     public GameObject hasilPlayer;
16.
17.     [HideInInspector]
18.     public int numberOfCombination;
19.     [HideInInspector]
20.     public string combInputString = "";
21.     [HideInInspector]
22.     public string correctnessString = "";
23.     [HideInInspector]
24.     public string[] combinationAns;
25.     [HideInInspector]
26.     public string[] combinationInput;

```



```

27.     public string[] correctness = new string[2] {
        "0", "0" };
28.     [HideInInspector]
29.     public int indexNumInput = 0;
30.     [HideInInspector]
31.     public int correctNumber = 0;
32.     [HideInInspector]
33.     public int correctPosition = 0;
34.
35.     void Start() {
36.         dataLoadSoal = soalLegot.GetComponent<LoadSoal>();
37.         legotChestController = chestLegot.GetComponent<ChestController>();
38.         printComponent = hasilPlayer.GetComponent<PrintJawabanPlayer>();
39.
40.         combinationAns = LegotDataContainer.LegotDataList[dataLoadSoal.legotNumber].jawaban.Split(
            '|');
41.         if (dataLoadSoal.legotNumber >= 0 && dataLoadSoal.legotNumber <= 7)
42.         {
43.             numberOfCombination = 3;
44.         }
45.         else
46.         {
47.             numberOfCombination = 4;
48.         }
49.
50.         combinationInput = new string[numberOfCombination];
51.         Array.Clear(combinationInput, 0, combinationInput.Length);
52.     }
53.
54.     void Update() {
55.         if (indexNumInput == numberOfCombination)
56.         {
57.             printComponent.goPrint = true;

```

```

58.         if (correctNumber == numberOfCombinat
ion && correctPosition == numberOfCombination)
59.         {
60.             legotChestController.flagPlayAnim
= true;
61.         }
62.         else
63.         {
64.             StartCoroutine(WaitAndClear());
65.         }
66.
67.         StartCoroutine(WaitAndVoid());
68.
69.     }
70. }
71.
72. IEnumerator WaitAndClear()
73. {
74.     yield return new WaitForSeconds(0.5f);
75.     Array.Clear(combinationInput, 0, combinat
ionInput.Length);
76. }
77.
78. IEnumerator WaitAndVoid()
79. {
80.     yield return new WaitForSeconds(0.5f);
81.     indexNumInput = 0;
82.     correctNumber = 0;
83.     correctPosition = 0;
84. }
85. }
86.

```

**Kode Sumber 4.3 Kelas LegotGameController**

#### 4.2.3.2 Data Controller

Objek Data Controller memiliki kelas LegotDataContainer. Kelas ini berfungsi untuk mengambil data dari sebuah file text yang menyimpan pertanyaan untuk *minigame* Legot. Data-data yang telah diambil akan disimpan pada sebuah array untuk

mempermudah pengambilan pertanyaan pada objek Pertanyaan. Objek ini tidak memiliki antarmuka karena berupa Game Object kosong yang menyimpan kelas LegotDataController. Kelas LegotDataContainer dapat dilihat pada Kode Sumber 4.4 di bawah.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using System.Xml;
5. using System.Xml.Serialization;
6. using System.IO;
7.
8. public class LegotDataContainer : MonoBehaviour{
9.
10.     public TextAsset LegotDatabasePath;
11.     public static List<LegotData> LegotDataList =
        new List<LegotData>();
12.
13.     private List<Dictionary<string, string>> lego
        tDataDictList = new List<Dictionary<string, strin
        g>>();
14.     private Dictionary<string, string> legotDataD
        ictionary;
15.
16.     private void Awake()
17.     {
18.         ReadLegotDatabase();
19.         for(int a = 0; a<legotDataDictList.Count;
                a++)
20.         {
21.             LegotDataList.Add(new LegotData(legot
                DataDictList[a]));
22.         }
23.     }
24.
25.     public void ReadLegotDatabase()
26.     {
27.         XmlDocument xmlDocument = new XmlDocument
            ();

```

```

28.
29.         xmlDocument.Load("F:/AVIAR (14_065,066,12
30.         4)/AVIAR/Assets/Resources/DataLegot2.xml");
31.         XmlNodeList ldList = xmlDocument.GetElemen
32.         tsByTagName("Item");
33.         foreach (XmlNode itemInfo in ldList)
34.         {
35.             XmlNodeList itemContent = itemInfo.Ch
36.             ildNodes;
37.             legotDataDictionary = new Dictionary<
38.             string, string>();
39.             foreach(XmlNode content in itemConten
40.             t)
41.             {
42.                 legotDataDictionary.Add(content.N
43.                 ame.ToString(), content.InnerText);
44.             }
45.             legotDataDictList.Add(legotDataDictio
46.             nary);
47.         }
48.     }

```

**Kode Sumber 4.4 Kelas LegotDataContainer**

#### **4.2.3.3 Legot Button**

Legot Button merupakan objek gabungan dari tombol – tombol angka. Masing – masing tombol angka memiliki kelas LegotInput. Kelas ini yang nantinya akan mengirim data angka yang dimasukkan oleh pemain. Data tersebut akan dikirimkan ke kelas LegotGameController untuk nantinya disimpan ke dalam array of string. Objek ini memiliki tag khusus yang akan

dibaca oleh objek Interaction. Kelas LegotInput bisa dilihat pada Kode Sumber 4.5 dibawah.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class LegotInput : MonoBehaviour {
6.
7.     private LegotGameController lgc;
8.     public GameObject legotMaster;
9.     public bool isActive = false;
10.
11.     void Start () {
12.         lgc = legotMaster.GetComponent<LegotGameC
ontroller>();
13.     }
14.
15.     private void Update()
16.     {
17.         if(isActive == true)
18.         {
19.             isActive = false;
20.             Selected();
21.         }
22.     }
23.
24.     private void Selected()
25.     {
26.         lgc.combinationInput[lgc.indexNumInput] =
gameObject.name;
27.         for(int a = 0; a < lgc.numberofCombinatio
n; a++)
28.         {
29.             if(lgc.combinationInput[lgc.indexNumI
nput] == lgc.combinationAns[a])
30.             {
31.                 lgc.correctNumber++;
32.             }
33.         }
34.

```

```

35.         if (lgc.combinationInput[lgc.indexNumInput
36.             t] == lgc.combinationAns[lgc.indexNumInput])
37.             {
38.                 lgc.correctPosition++;
39.             }
40.         //Debug.Log(correctNumber + " " + correct
41.             Position);
42.         lgc.correctness[0] = lgc.correctNumber.To
43.             ToString();
44.         lgc.correctness[0] += "A,";
45.         lgc.correctness[1] = lgc.correctPosition.
46.             ToString();
47.         lgc.correctness[1] += "P ";
48.         //Debug.Log("index: " + indexNumInput + "
49.             | " + "(0)" + combinationInput[0] + "(1)" + comb
50.             inationInput[1] + "(2)" + combinationInput[2]);
51.         if (lgc.indexNumInput == lgc.numberOfComb
52.             ination-1)
53.             {
54.                 lgc.combInputString = string.Join(",",
55.                     lgc.combinationInput);
56.                 lgc.correctnessString = string.Join("
57.                     ", lgc.correctness);
58.                 //Debug.Log(combInputString);
59.             }
60.         lgc.indexNumInput += 1;
61.     }
62. }

```

**Kode Sumber 4.5 Kelas LegotInput**

Selain tombol angka, terdapat juga tombol *restart*. Tombol ini digunakan untuk menghapus jawaban yang belum masuk ke perhitungan. Maksudnya adalah ketika pemain baru memasukkan satu atau dua angka maka jawaban bisa dihapus untuk menjawab

ulang. Hal ini dilakukan agar pemain tidak terburu-buru dalam menjawab pertanyaan. Kelas LegotRestartAnswer menjadi kelas yang akan menjalankan fungsi tersebut. Kelas LegotRestartAnswer dapat dilihat pada Kode Sumber 4.6 dibawah.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using System;
5.
6. public class LegotRestartAnswer : MonoBehaviour {
7.
8.     private LegotGameController lgc;
9.     private PrintKombinasiJawaban pkj;
10.
11.     public GameObject legotMaster;
12.     public GameObject answer1;
13.     public GameObject answer2;
14.     public GameObject answer3;
15.     public GameObject answer4;
16.     public bool activated;
17.
18.     private void Start()
19.     {
20.         activated = false;
21.         lgc = legotMaster.GetComponent<LegotGameC
ontroller>();
22.     }
23.
24.     void Update () {
25.         if(activated == true)
26.         {
27.             activated = false;
28.             Array.Clear(lgc.combinationInput, 0,
lgc.combinationInput.Length);
29.             lgc.correctnessString = "";
30.             lgc.combInputString = "";
31.             lgc.indexNumInput = 0;
32.             lgc.correctNumber = 0;
33.             lgc.correctPosition = 0;

```

```

34.
35.         pkj = answer1.GetComponent<PrintKombi
        nasiJawaban>();
36.         pkj.number = "";
37.         pkj = answer2.GetComponent<PrintKombi
        nasiJawaban>();
38.         pkj.number = "";
39.         pkj = answer3.GetComponent<PrintKombi
        nasiJawaban>();
40.         pkj.number = "";
41.         pkj = answer4.GetComponent<PrintKombi
        nasiJawaban>();
42.         pkj.number = "";
43.     }
44. }
45. }

```

**Kode Sumber 4.6 Kelas LegotRestartAnswer**

#### 4.2.3.4 Soal Legot

Objek ini merupakan objek yang akan menampilkan pertanyaan dari sebuah *minigame* Legot. Objek ini memiliki kelas LoadSoal. Pertanyaan akan diambil dari kelas LegotDataContainer. Pertanyaan yang akan ditampilkan ditentukan secara manual. Kelas LoadSoal dapat dilihat pada Kode Sumber 4.7.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class LoadSoal : MonoBehaviour {
6.
7.     public int legotNumber;
8.
9.     void Start () {
10.         GetComponent<TextMesh>().text = LegotData
        Container.LegotDataList[legotNumber].soal;
11.

```



```

12.     }
13. }

```

#### Kode Sumber 4.7 Kelas LoadSoal

#### 4.2.3.5 Jawaban Player

Objek ini merupakan objek yang akan menampilkan jawaban player beserta status kebenarannya. Objek ini terdiri dari dua bagian, yaitu bagian yang menampilkan angka dan status kebenaran.

Bagian yang menunjukkan angka memiliki kelas PrintKombinasiJawaban. Kelas ini akan menampilkan angka sesuai dengan urutan ketika pemain sedang menjawab. Jadi apabila pemain sedang menjawab angka pertama maka akan muncul angka pertama dan begitu hingga angka terakhir. Hal ini dilakukan agar pemain tahu di urutan keberapa pemain sedang menjawab. Kelas PrintKombinasiJawaban dapat dilihat pada Kode Sumber 4.8.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class PrintKombinasiJawaban : MonoBehaviour
6. {
7.     public GameObject legotMaster;
8.     public int digitKe;
9.
10.    private LegotGameController legotMasterComponent;
11.
12.    private void Start()
13.    {
14.        legotMasterComponent = legotMaster.GetComponent<LegotGameController>();
15.    }
16.

```

```

17.     void Update () {
18.         changeNumber();
19.     }
20.
21.     void changeNumber()
22.     {
23.         if (digitKe != 1)
24.         {
25.             if (legotMasterComponent.indexNumInput
26. t == 1)
27.             {
28.                 GetComponent<TextMesh>().text = "
29. ";
30.             }
31.             else
32.             {
33.                 GetComponent<TextMesh>().text = legot
34. MasterComponent.combinationInput[(digitKe - 1
35. )].ToString();
36.                 Debug.Log("angka ke " + digitKe +
37. " print " + legotMasterComponent.combinationInpu
38. t[(digitKe - 1)].ToString());
39.             }
40.         }

```

**Kode Sumber 4.8 Kelas PrintKombinasiJawaban**

Lalu bagian yang memiliki status kebenaran memiliki kelas PrintJawabanPlayer. Kelas ini bertugas untuk menampilkan status kebenaran dari jawaban pemain. Kelas PrintJawabanPlayer dapat dilihat pada Kode Sumber 4.9.

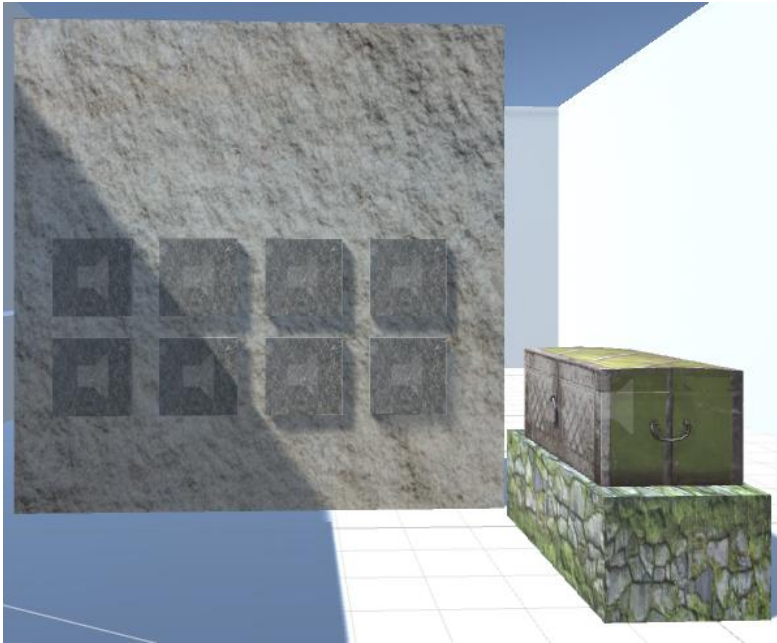
```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class PrintJawabanPlayer : MonoBehaviour {
6.
7.     private LegotGameController lgc;
8.
9.     public bool goPrint = false;
10.    public GameObject legotMaster;
11.
12.    private void Start()
13.    {
14.        lgc = legotMaster.GetComponent<LegotGameC
ontroller>();
15.    }
16.
17.    void Update () {
18.
19.        if (lgc.indexNumInput == 1)
20.        {
21.            GetComponent<TextMesh>().text = "";
22.        }
23.
24.        if(goPrint == true)
25.        {
26.            goPrint = false;
27.            GetComponent<TextMesh>().text = " = "
+ lgc.correctnessString;
28.        }
29.    }
30. }

```

**Kode Sumber 4.9 Kelas PrintJawabanPlayer**

#### 4.2.4 Picture Memory



**Gambar 4.2 Implementasi Picture Memory**

Gambar 4.2 merupakan gambar dari implementasi *minigame* Picture Memory. Pada *minigame* ini terdapat tiga objek dengan kelasnya masing – masing yang saling berkomunikasi. Objek tersebut adalah objek *minigame* secara keseluruhan yaitu Picture Memory, Picture Memory Buttons, dan Picture Button.

##### 4.2.4.1 Picture Memory

Objek Picture Memory merupakan objek *minigame* secara keseluruhan. Objek ini memiliki kelas PictureMemoryMaster yang berfungsi sebagai pengatur jalannya permainan. Pertama objek ini menyimpan tingkat kesulitan dari *minigame* tersebut. Lalu objek ini

juga yang bertanggung jawab dalam mencocokkan gambar pada panel yang dipilih oleh pemain dengan menerima string dari nama panel yang dikirimkan oleh kelas PictureMemoryController. Ketika ada dua panel dipilih untuk dicocokkan maka akan terjadi jeda dimana panel yang lain menjadi tidak bisa dipilih. Objek ini bertanggung jawab untuk memberi sinyal kepada panel yang tidak dipilih agar tidak berputar walaupun panel tersebut ditekan selama durasi jeda. Kelas PictureMemoryMaster dapat dilihat pada Kode Sumber 4.10.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using System;
5.
6. public class PictureMemoryMaster : MonoBehaviour
7. {
8.
9.     private ChestController pmChestController;
10.
11.     public int difficulty; //easy: 2, med: 3, hard
    : 4
12.     public GameObject chest;
13.     public GameObject pmButtonsGroup;
14.
15.     [HideInInspector]
16.     public int pointToWin;
17.     [HideInInspector]
18.     public int pointCollected;
19.     [HideInInspector]
20.     public int indexOfChoice;
21.     [HideInInspector]
22.     public string[] compareChoice = new string[2]
    { "0", "0" };
23.     [HideInInspector]
24.     public string[] realName = new string[2] { "0"
    , "0" };
25.

```

```

26.     void Start()
27.     {
28.         pmChestController = chest.GetComponent<Che
stController>();
29.         pointToWin = difficulty * 2;
30.         indexOfChoice = 0;
31.         pointCollected = 0;
32.     }
33.
34.     void Update()
35.     {
36.         if (indexOfChoice == 2)
37.         {
38.             Transform[] childObject = new Transfor
m[2];
39.             PictureMemoryController[] coComponent
= new PictureMemoryController[2];
40.
41.             childObject[0] = gameObject.transform.
Find("PictureButtons/" + realName[0]);
42.             coComponent[0] = childObject[0].GetCom
ponent<PictureMemoryController>();
43.             childObject[1] = gameObject.transform.
Find("PictureButtons/" + realName[1]);
44.             coComponent[1] = childObject[1].GetCom
ponent<PictureMemoryController>();
45.
46.             indexOfChoice = 0;
47.             StartCoroutine(giveSignal(coComponent[
0], coComponent[1]));
48.         }
49.
50.         if (pointCollected == pointToWin)
51.         {
52.             Debug.Log("Chest unlocked");
53.             pmChestController.flagPlayAnim = true;
54.
55.             Debug.Log(pmChestController.flagPlayAn
im);
56.             pointCollected = 0;
57.         }
58.     }

```

```

58.
59.     IEnumerator giveSignal(PictureMemoryController
        coComponent1, PictureMemoryController coComponent
        2)
60.     {
61.         PictureMemoryController buttonComponent;
62.
63.         //disabling other button so that it cant b
        e clicked while waiting for result
64.         for (int a = 0; a < difficulty * 4; a++)
65.         {
66.             buttonComponent = pmButtonsGroup.trans
                form.GetChild(a).GetComponent<PictureMemoryControl
                ler>());
67.             buttonComponent.flipped = true;
68.         }
69.
70.         yield return new WaitForSeconds(2.0f);
71.         Debug.Log(compareChoice[0] + " " + compare
                Choice[1]);
72.
73.         //enabling the button to be clicked
74.         for (int a = 0; a < difficulty * 4; a++)
75.         {
76.             buttonComponent = pmButtonsGroup.trans
                form.GetChild(a).GetComponent<PictureMemoryControl
                ler>());
77.             buttonComponent.flipped = false;
78.         }
79.
80.         if (compareChoice[0] == compareChoice[1])
81.         {
82.             coComponent1.signalRight = true;
83.             coComponent2.signalRight = true;
84.
85.             pointCollected += 1;
86.             Debug.Log("pointCollected: " + pointCo
                llected);
87.         }
88.         else
89.         {

```

```

90.         coComponent1.signalWrong = true;
91.         coComponent2.signalWrong = true;
92.     }
93.
94.     Array.Clear(compareChoice, 0, compareChoice.Length);
95. }
96. }

```

**Kode Sumber 4.10 Kelas PictureMemoryMaster**

#### 4.2.4.2 Picture Buttons

Objek Picture Buttons merupakan objek kumpulan dari panel – panel pada *minigame* Picture Memory. Objek ini memiliki kelas PictureMemoryButtonPos yang berfungsi untuk mengacak posisi panel – panel yang ada sehingga tiap permainan posisi dari gambar tidak selalu sama. Tiap tingkat kesulitan pada *minigame* ini memiliki jumlah panel yang berbeda. Maka dari itu kelas ini berfungsi untuk menonaktifkan panel yang tidak terpakai di tingkat kesulitan yang rendah. Karena dari awal, panel yang tersedia berjumlah enam belas, yaitu jumlah maksimal panel. Kelas PictureMemoryButtonPos dapat dilihat pada Kode Sumber 4.11.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using System;
5.
6. public class PictureMemoryButtonPos : MonoBehaviour
7. {
8.     private int randomNumber;
9.     private int[] pictSpawnPosFlag;
10.    private Vector3[] pictButtonPos;
11.    private Vector3 assignedPosition;
12.    private PictureMemoryMaster pmMasterComponent;

```



```

13.
14.     //Easy: 8, Med: 12, Hard: 16
15.     public GameObject pict11;
16. public GameObject pict12;
17.     public GameObject pict21;
18. public GameObject pict22;
19.     public GameObject pict31; public GameObject pi
    ct32;
20.     public GameObject pict41; public GameObject pi
    ct42;
21.     public GameObject pict51; public GameObject pi
    ct52;
22.     public GameObject pict61; public GameObject pi
    ct62;
23.     public GameObject pict71; public GameObject pi
    ct72;
24.     public GameObject pict81; public GameObject pi
    ct82;
25.
26.     public GameObject pmMaster;
27.
28.     void Start()
29.     {
30.         pmMasteComponent = pmMaster.GetComponent<P
    ictureMemoryMaster>();
31.         pictSpawnPosFlag = new int[pmMasteComponen
    t.difficulty * 4];
32.         pictButtonPos = new Vector3[pmMasteCompone
    nt.difficulty * 4];
33.         Array.Clear(pictSpawnPosFlag, 0, pictSpawn
    PosFlag.Length);
34.
35.         //inserting position
36.         for (int a = 0; a < pmMasteComponent.diffi
    culty; a++)
37.         {
38.             for (int b = 0; b < 4; b++)
39.             {
40.                 pictButtonPos[((a * 4) + b)] = new
    Vector3((1.7f * b), (1.6f * a), 0.1f);
41.             }
42.         }

```

```

43.
44.     //placing picture randomly according to the
    position assigned in the previous loop
45.     randomNumber = randomNumberGenerator(); placePicture(pict11, randomNumber);
46.     randomNumber = randomNumberGenerator(); placePicture(pict12, randomNumber);
47.     randomNumber = randomNumberGenerator(); placePicture(pict21, randomNumber);
48.     randomNumber = randomNumberGenerator(); placePicture(pict22, randomNumber);
49.     randomNumber = randomNumberGenerator(); placePicture(pict31, randomNumber);
50.     randomNumber = randomNumberGenerator(); placePicture(pict32, randomNumber);
51.     randomNumber = randomNumberGenerator(); placePicture(pict41, randomNumber);
52.     randomNumber = randomNumberGenerator(); placePicture(pict42, randomNumber);
53.
54.     if (pmMasteComponent.difficulty == 3 || pm
    MasteComponent.difficulty == 4)
55.     {
56.         randomNumber = randomNumberGenerator()
    ; placePicture(pict51, randomNumber);
57.         randomNumber = randomNumberGenerator()
    ; placePicture(pict52, randomNumber);
58.         randomNumber = randomNumberGenerator()
    ; placePicture(pict61, randomNumber);
59.         randomNumber = randomNumberGenerator()
    ; placePicture(pict62, randomNumber);
60.
61.     }
62.
63.     if (pmMasteComponent.difficulty == 4)
64.     {
65.         randomNumber = randomNumberGenerator()
    ; placePicture(pict71, randomNumber);
66.         randomNumber = randomNumberGenerator()
    ; placePicture(pict72, randomNumber);
67.         randomNumber = randomNumberGenerator()
    ; placePicture(pict81, randomNumber);

```

```

68.         randomNumber = randomNumberGenerator()
        ; placePicture(pict82, randomNumber);
69.
70.     }
71.
72.     //destroy unused picture for easy and med
73.     if(pmMasteComponent.difficulty == 2 || pmM
asteComponent.difficulty == 3)
74.     {
75.         pict81.SetActive(false);
76.         pict82.SetActive(false);
77.         pict71.SetActive(false);
78.         pict72.SetActive(false);
79.     }
80.
81.     if (pmMasteComponent.difficulty == 2)
82.     {
83.         pict61.SetActive(false);
84.         pict62.SetActive(false);
85.         pict51.SetActive(false);
86.         pict52.SetActive(false);
87.     }
88. }
89.
90. public int randomNumberGenerator()
91. {
92.     int randomNumberTemp;
93.
94.     randomNumberTemp = UnityEngine.Random.Rang
e(0, pmMasteComponent.difficulty * 4);
95.     while (pictSpawnPosFlag[randomNumberTemp]
== 1)
96.     {
97.         randomNumberTemp = UnityEngine.Random.
Range(0, pmMasteComponent.difficulty * 4);
98.     }
99.
100.    pictSpawnPosFlag[randomNumberTemp]
= 1;
101.    return randomNumberTemp;
102. }

```

```

103.
104.         private void placePicture(GameObject pi
           ctureToBePlaced, int indexOfPosition)
105.         {
106.             pictureToBePlaced.transform.localPo
           sition = pictButtonPos[indexOfPosition];
107.         }
108.     }

```

**Kode Sumber 4.11 Kelas PictureMemoryButtonPos**

#### 4.2.4.3 Picture Button

Objek Picture Button adalah objek panel yang menyimpan gambar yang juga menjadi anak dari objek Picture Buttons. Objek ini memiliki kelas PictureMemoryController yang berfungsi untuk memutar panel ketika ditekan. Selain itu kelas ini juga yang mengirim nama dari panel yang ditekan ke PictureMemoryMaster yang nantinya akan dibandingkan. Objek ini memiliki tag khusus yang akan dibaca oleh objek Interaction. Kelas PictureMemoryController dapat dilihat pada Kode Sumber 4.12.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using System;
5. using UnityEngine.Events;
6.
7. public class PictureMemoryController : MonoBehaviour {
8.
9.     private PictureMemoryMaster pmMasterComponent;
10.
11.     private bool flippingToPicture = false;
12.     private bool flippingToBack = false;
13.     private bool pickedThisRound = false;

```

```

13.     private bool canBeClicked = false;
14.     private float smooth = 3;
15.     private Quaternion initialRotation;
16.     private Quaternion targetRotation;
17.
18.     public bool flipped = false;
19.     public bool signalRight = false;
20.     public bool signalWrong = false;
21.     public GameObject pmMaster;
22.     public bool isActive = false;
23.
24.     private void Awake()
25.     {
26.         targetRotation = gameObject.transform.rotation;
27.     }
28.
29.     void Start () {
30.         pmMasterComponent = pmMaster.GetComponent<
31.         PictureMemoryMaster>();
32.         initialRotation.eulerAngles = new Vector3(
33.         0.0f, 0.0f, 0.0f);
34.         targetRotation.eulerAngles = new Vector3(0
35.         .0f, 180.0f, 0.0f);
36.         canBeClicked = true;
37.     }
38.
39.     void Update()
40.     {
41.         if(isActive == true)
42.         {
43.             isActive = false;
44.             Selected();
45.         }
46.
47.         Quaternion currentRotation = gameObject.tr
48.         ansform.rotation;
49.         if(pickedThisRound == true)
50.         {
51.             if (signalWrong == true)
52.             {
53.                 //Debug.Log("masuk !=");

```

```

50.         flipped = false;
51.         signalWrong = false;
52.         pickedThisRound = false;
53.         flippingToPicture = false;
54.         flippingToBack = true;
55.         canBeClicked = false;
56.         flipPicture();
57.     }
58.     else if (signalRight == true)
59.     {
60.         //Debug.Log("masuk ==");
61.         flipped = true;
62.         signalRight = false;
63.         pickedThisRound = false;
64.     }
65. }
66.
67.     flipPicture();
68.     if(gameObject.transform.rotation == initialRotation) canBeClicked = true;
69.
70.     if(gameObject.transform.rotation == initialRotation || gameObject.transform.rotation == targetRotation)
71.     {
72.         flippingToBack = false;
73.         flippingToPicture = false;
74.     }
75. }
76.
77. private void Selected()
78. {
79.     string temps = gameObject.name;
80.     temps = temps.TrimEnd(temps[temps.Length-
81. 1]);
82.     if(canBeClicked == true)
83.     {
84.         if (flipped == false)
85.         {
86.             pmMasterComponent.compareChoice[pm
MasterComponent.indexOfChoice] = temps;

```

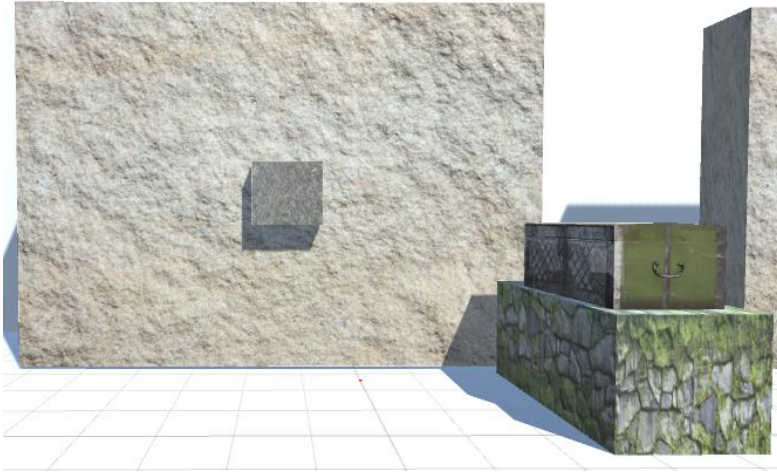
```

87.         pmMasterComponent.realName[pmMaste
rComponent.indexOfChoice] = gameObject.name;
88.         flipped = true;
89.         flippingToPicture = true;
90.         pickedThisRound = true;
91.
92.         pmMasterComponent.indexOfChoice =
pmMasterComponent.indexOfChoice + 1;
93.     }
94. }
95.
96. }
97.
98. public void flipPicture()
99. {
100.     canBeClicked = false;
101.
102.     if (flippingToPicture == true)
103.     {
104.         gameObject.transform.rotation =
Quaternion.Slerp(transform.rotation, targetRotati
on, Time.deltaTime * smooth);
105.     }
106.
107.     if(flippingToBack == true)
108.     {
109.         gameObject.transform.rotation =
Quaternion.Slerp(transform.rotation, initialRotati
on, Time.deltaTime * smooth);
110.     }
111. }
112. }

```

**Kode Sumber 4.12 Kelas PictureMemoryController**

### 4.2.5 Tap Fast



**Gambar 4.3 Implementasi Tap Fast**

Gambar 4.3 diatas merupakan hasil implementasi *minigame* Tap Fast. Pada *minigame* ini terdapat tiga objek yang saling berkomunikasi. Objek tersebut adalah objek *minigame* secara keseluruhan yaitu Tap Fast, Start Button, dan Button.

#### 4.2.5.1 Tap Fast

Objek Tap Fast merupakan objek *minigame* secara keseluruhan. Objek ini memiliki kelas TapFastController yang berfungsi untuk memulai permainan ketika objek Start Button ditekan. Ketika permainan dimulai, kelas TapFastController juga yang akan menginisiasi ronde. Pada tiap rondanya, kelas ini akan memberi tahu kepada objek Buttons silinder mana yang harus digerakkan. Kelas TapFastController dapat dilihat pada Kode Sumber 4.13.

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
```



```

4.
5. public class TapFastController : MonoBehaviour {
6.
7.     private int pointsToWin;
8.     private int numberOfButtons;
9.     private ChestController chestController;
10.
11.     public int difficulty; //Easy = 1, Med = 2 Ha
    rd = 3
12.     [HideInInspector]
13.     public int pointsCollected = 0;
14.     [HideInInspector]
15.     public bool isPlaying = false;
16.     [HideInInspector]
17.     public bool haltForRound = false;
18.     [HideInInspector]
19.     public bool haltForMove = false;
20.     [HideInInspector]
21.     public int minButtonMoved;
22.     [HideInInspector]
23.     public int maxButtonsMoved;
24.     [HideInInspector]
25.     public int numberOfRounds;
26.     [HideInInspector]
27.     public bool startFlag = false;
28.
29.     public GameObject chest;
30.     public List<GameObject> tfButtons = new List<
    GameObject>();
31.
32.     void Awake () {
33.         chestController = chest.GetComponent<Ches
    tController>();
34.         numberOfButtons = difficulty * 7-
    (2 * (difficulty-1));
35.         numberOfRounds = difficulty + 5-(1-
    difficulty);
36.
37.         switch (difficulty)
38.         {
39.             case 1:

```

```

40.             minButtonMoved = 1;
41.             maxButtonsMoved = 3;
42.             break;
43.         case 2:
44.             minButtonMoved = 2;
45.             maxButtonsMoved = 4;
46.             break;
47.         case 3:
48.             minButtonMoved = 2;
49.             maxButtonsMoved = 6;
50.             break;
51.     }
52.
53.     pointsToWin = numberOfRounds * minButtonM
oved;
54. }
55.
56. private void Update()
57. {
58.     if(startFlag == true)
59.     {
60.         startFlag = false;
61.         initiateGame();
62.     }
63.
64.     if (pointsCollected >= pointsToWin)
65.     {
66.         chestController.flagPlayAnim = true;
67.     }
68. }
69.
70. public void initiateGame()
71. {
72.     isPlaying = true;
73.     StartCoroutine(StartGame());
74. }
75.
76. IEnumerator StartGame()
77. {
78.     for (int a = 0; a < numberOfRounds; a++)

```

```

79.         {
80.             int numberOfButtonsMoved = Random.Ran
ge(minButtonMoved, maxButtonsMoved);
81.             haltForRound = true;
82.             StartCoroutine(InitiateRound(numberOf
ButtonsMoved));
83.
84.             yield return new WaitUntil(() => halt
ForRound == false);
85.         }
86.
87.         if (pointsCollected <= pointsToWin)
88.         {
89.             pointsCollected = 0;
90.             isPlaying = false;
91.         }
92.     }
93.
94.     IEnumerator InitiateRound(int buttonMoved)
95.     {
96.         for(int a = 0; a < buttonMoved; a++)
97.         {
98.             int buttonIndexMoved = Random.Range(0
, numberOfButtons-1);
99.             GameObject buttonToMove = tfButtons[b
uttonIndexMoved];
100.             TapFastButton buttonComponent
= buttonToMove.GetComponent<TapFastButton>();
101.             buttonComponent.canBeClicked =
true;
102.             buttonComponent.signalMove = t
rue;
103.             buttonComponent.movedThisRound
= true;
104.             haltForMove = true;
105.
106.             if (a == buttonMoved-1)
107.             {
108.                 yield return new WaitUntil
(() => haltForMove == false);
109.             }
110.         }

```

```

111.
112.             haltForRound = false;
113.         }
114.     }

```

**Kode Sumber 4.13 Kelas TapFastController**

#### 4.2.5.2 Start Button

Objek Start Button merupakan objek tombol yang jika ditekan maka akan menginisiasi permainan. Objek ini memiliki kelas TapFastStart yang berfungsi untuk mengirimkan sinyal kepada kelas TapFastController untuk memulai permainan ketika objek ini ditekan. Kelas TapFastStart bisa dilihat pada Kode Sumber 4.14.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.Events;
5.
6. public class TapFastStart : MonoBehaviour {
7.
8.     private TapFastController tfMasterComponent;
9.
10.    private Vector3 originalPosition;
11.
12.    public GameObject tfMaster;
13.    public bool isActive = false;
14.
15.    void Start () {
16.        originalPosition = gameObject.transform.localPosition;
17.        tfMasterComponent = tfMaster.GetComponent<TapFastController>();
18.    }
19.
20.    private void Update()
21.    {
22.        if(isActive == true)

```

```

23.         isActive = false;
24.         tfMasterComponent.startFlag = true;
25.     }
26.
27.     if(tfMasterComponent.isPlaying == true)
28.     {
29.         gameObject.transform.localPosition =
new Vector3(gameObject.transform.localPosition.x,
gameObject.transform.localPosition.y, 0f);
30.     }
31.     else
32.     {
33.         gameObject.transform.localPosition =
originalPosition;
34.     }
35. }
36. }

```

**Kode Sumber 4.14 Kelas TapFastStart**

#### 4.2.5.3 Button

Objek Buttons merupakan kumpulan silinder – silinder yang harus ditekan oleh pemain ketika berada di luar tembok. Silinder pada objek ini memiliki kelas TapFastButton yang berfungsi untuk menggerakkan silinder ketika telah diberikan sinyal oleh TapFastController. Objek ini memiliki tag khusus yang akan dibaca oleh objek Interaction. Kelas TapFastController dapat dilihat pada Kode Sumber 4.15.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class TapFastButton : MonoBehaviour {
6.
7.     private TapFastController tfMasterComponent;
8.
9.     private Vector3 newPosition;
9.     private Vector3 insidePosition;
10.    private Vector3 outsidePosition;

```

```

11.     private float timePassed;
12.     private float smooth;
13.     private bool moveForward = false;
14.     private bool moveBackward = false;
15.
16.     public GameObject tfMaster;
17.     public bool signalMove = false;
18.     public bool movedThisRound = false;
19.     public bool isClicked = false;
20.     public bool canBeClicked = false;
21.
22.
23.     private void Awake()
24.     {
25.         newPosition = transform.position;
26.     }
27.
28.     void Start () {
29.         tfMasterComponent = tfMaster.GetComponent
<TapFastController>();
30.         insidePosition = gameObject.transform.localPosition;
31.         outsidePosition = new Vector3(gameObject.transform.localPosition.x, gameObject.transform.localPosition.y, -0.70f);
32.
33.         switch (tfMasterComponent.difficulty)
34.         {
35.             case 1:
36.                 smooth = 2.5f;
37.                 break;
38.             case 2:
39.                 smooth = 5;
40.                 break;
41.             case 3:
42.                 smooth = 7;
43.                 break;
44.         }
45.     }
46.
47.     private void Update()
48.     {

```

```

49.         if(canBeClicked == true)
50.         {
51.             if (isClicked == true)
52.             {
53.                 isClicked = false;
54.                 canBeClicked = false;
55.                 if (tfMasterComponent.isPlaying =
= true)
56.                 {
57.                     tfMasterComponent.pointsColle
cted += 1;
58.                 }
59.             }
60.         }
61.
62.         if(signalMove == true)
63.         {
64.             signalMove = false;
65.             moveForward = true;
66.         }
67.
68.         moveButton();
69.
70.         if (gameObject.transform.localPosition ==
outsidePosition)
71.         {
72.             moveForward = false;
73.             moveBackward = true;
74.         }
75.
76.         if (gameObject.transform.localPosition ==
insidePosition)
77.         {
78.             moveForward = false;
79.             moveBackward = false;
80.         }
81.
82.         if(movedThisRound == true)
83.         {
84.             if (moveBackward == false && moveForw
ard == false)
85.             {

```

```

86.         tfMasterComponent.haltForMove = f
    else;
87.         movedThisRound = false;
88.         canBeClicked = false;
89.     }
90. }
91. }
92.
93. void moveButton()
94. {
95.     if(moveForward == true)
96.     {
97.         gameObject.transform.localPosition =
Vector3.Lerp(transform.localPosition, outsidePosi
tion, Time.deltaTime * smooth);
98.     }
99.
100.    if(moveBackward == true)
101.    {
102.        gameObject.transform.localPosi
tion = Vector3.Lerp(transform.localPosition, insi
dePosition, Time.deltaTime * smooth);
103.    }
104. }
105. }

```

**Kode Sumber 4.15 Kelas TapFastButton**

#### 4.2.5.4 Tap Fast Text

Objek Tap Fast Text merupakan objek yang memunculkan petunjuk bermain, jumlah minimal silinder yang harus ditekan untuk menang, dan jumlah pilar yang berhasil ditekan oleh pemain. Objek ini memiliki kelas TapFastText yang dapat dilihat pada Kode Sumber 4.16.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class TapFastText : MonoBehaviour {

```



```

6.
7.     private TapFastController tfController;
8.     public GameObject tfMaster;
9.
10.    void Start () {
11.        tfController = tfMaster.GetComponent<TapFastController>();
12.    }
13.
14.    private void Update()
15.    {
16.        GetComponent<TextMesh>().text = "Press the pillar that came out. Press " + (tfController.minButtonMoved * tfController.numberOfRounds) + " times\nto open chest. Press the square below to start\ncurrently pressed: " + tfController.pointsCollected;
17.    }
18.
19. }

```

**Kode Sumber 4.16 Kelas TapFastText**

#### 4.2.6 Chest

Chest adalah objek peti yang menyimpan senjata untuk nantinya bisa digunakan oleh pemain dalam melawan musuh. Objek ini sendiri terletak pada tiap-tiap *minigame*. Objek ini memiliki kelas ChestController yang berfungsi untuk menerima sinyal dari *minigame* ketika pemain berhasil menyelesaikan *minigame* tersebut. Ketika sinyal tersebut diterima oleh objek chest, maka objek ini akan memainkan animasi membuka peti agar pemain dapat mengambil senjata didalamnya. Kelas ChestController dapat dilihat pada Kode Sumber 4.17.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class ChestController : MonoBehaviour {

```

```

6.
7.     public GameObject minigameMaster;
8.     public Animation animasi;
9.     public bool opened = false;
10.    public bool flagPlayAnim = false;
11.
12.    void Update () {
13.        if (flagPlayAnim == true && opened == false)
14.        {
15.            playChestAnimation();
16.            if(gameObject.tag == "Centered")
17.            {
18.                StartCoroutine(WaitAndDestroy(3.5f));
19.                Destroy(minigameMaster);
20.            }
21.        }
22.    }
23.
24.    void playChestAnimation()
25.    {
26.        animasi.Play();
27.        flagPlayAnim = false;
28.        opened = true;
29.    }
30.
31.    IEnumerator WaitAndDestroy(float time)
32.    {
33.        yield return new WaitForSeconds(3.5f);
34.        Destroy(minigameMaster);
35.    }
36. }

```

**Kode Sumber 4.17 Kelas ChestController**

## **BAB V**

### **UJI COBA DAN EVALUASI**

Pada bab ini akan dijelaskan mengenai rangkaian uji coba dan evaluasi yang dilakukan. Proses pengujian dilakukan menggunakan metode *blackbox* berdasarkan skenario yang telah ditentukan.

#### **5.1 Lingkungan Uji Coba**

Lingkungan implementasi dari tugas akhir ini adalah sebagai berikut:

**Tabel 5.1 Spesifikasi lingkungan uji coba**

<b>Perangkat</b>	<b>Spesifikasi</b>
Perangkat Keras	<ul style="list-style-type: none"><li>• Prosesor Intel(R) Core (TM) i7-770 CPU @ 3.60GHz (8 CPUs), ~3.6 GHz</li><li>• RAM 8 Gigabyte</li><li>• Graphic card NVIDIA GeForce GTX 1060 3GB</li><li>• Oculus Rift</li></ul>
Perangkat Lunak	<ul style="list-style-type: none"><li>• Sistem operasi Windows 10 Home Single Language 64-bit</li><li>• Unity 2017.3.0f3 Personal (64-bit)</li></ul>

##### **5.1.1 Pengujian Fungsionalitas**

Pada sub bab ini akan dibahas hasil pengujian kebutuhan fungsionalitas yang mengacu kepada sub bab 3.2. Pengujian fungsionalitas dilakukan dengan menyiapkan sejumlah skenario. Penjelasan mengenai pengujian disajikan dengan menampilkan kondisi awal, masukan, keluaran yang diharapkan, hasil uji coba, dan kondisi akhir. Pengujian fungsionalitas dijabarkan sebagai berikut.

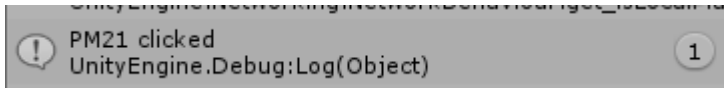
### 5.1.2 Uji Coba Kontroler

Pada pengujian kontroler dilakukan uji coba dengan mengarahkan laser pada objek tertentu, lalu menekan tombol trigger. Uji coba dikatakan berhasil apabila nama dari objek yang ditekan muncul pada kolom Debug pada unity. Skenario yang telah diuji dapat dilihat pada Tabel 5.2.

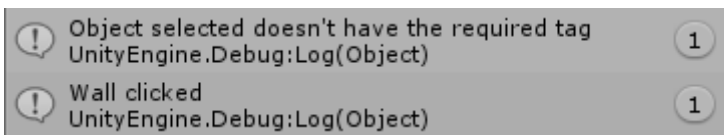
**Tabel 5.2 Skenario Uji Coba Kontroler**

ID	UF-001
Nama	Uji coba pada kontroler.
Tujuan uji coba	Pengguna mengetahui kontroler dapat digunakan untuk memilih objek pada <i>minigame</i> .
Kondisi awal	Pemain berada di depan <i>minigame</i> , kolom Debug pada unity kosong.
<b>Skenario 1</b>	<b><i>Pemain memilih objek yang dapat berinteraksi pada minigame.</i></b>
Masukkan	Pemain memilih objek dengan mengarahkan laser ke tombol yang ada pada <i>minigame</i> lalu menekan tombol trigger pada oculus touch.
Keluaran yang diharapkan	Kolom Debug pada unity menampilkan nama objek yang dipilih.
Hasil uji coba	Berhasil.
Kondisi akhir	Pada kolom Debug tertulis nama objek yang dipilih.
<b>Skenario 2</b>	<b><i>Pemain memilih objek yang tidak dapat berinteraksi pada minigame.</i></b>
Masukkan	Pemain memilih objek dengan mengarahkan laser ke objek selain tombol pada <i>minigame</i> lalu menekan tombol trigger pada oculus touch.
Keluaran yang diharapkan	Kolom Debug pada unity menampilkan nama objek yang dipilih dan penjelasan bahwa objek yang dipilih bukan objek tombol.
Hasil uji coba	Berhasil

Kondisi akhir	Pada kolom Debug tertulis nama objek yang dipilih dan penjelasan bahwa objek yang dipilih bukan objek tombol.
---------------	---



**Gambar 5.1 Hasil uji coba UF-001 Skenario 1**



**Gambar 5.2 Hasil uji coba UF-001 Skenario 2**

Hasil dari skenario 1 yaitu munculnya nama objek pada kolom Debug dapat dilihat pada Gambar 5.1. Lalu hasil skenario 2 yaitu memunculkan keterangan bahwa objek yang dipilih tidak memiliki tag dari tombol sehingga tidak terjadi interaksi dan nama dari objek tersebut tetap ditampilkan. Hasil skenario 2 dapat dilihat pada Gambar 5.2.

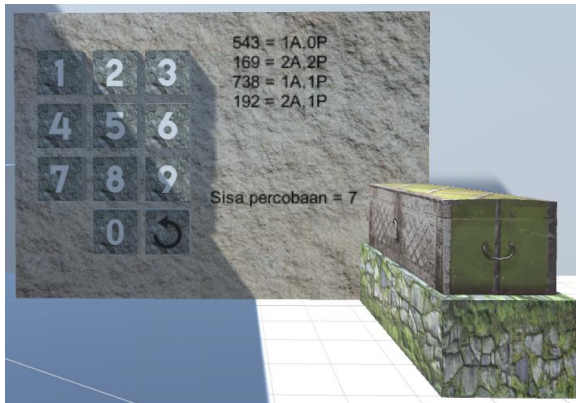
### 5.1.3 Uji Coba *Minigame* Legot

Pada pengujian *minigame* Legot dilakukan uji coba dengan memainkan *minigame* Legot secara langsung. Skenario yang telah diuji dapat dilihat pada Tabel 5.3.

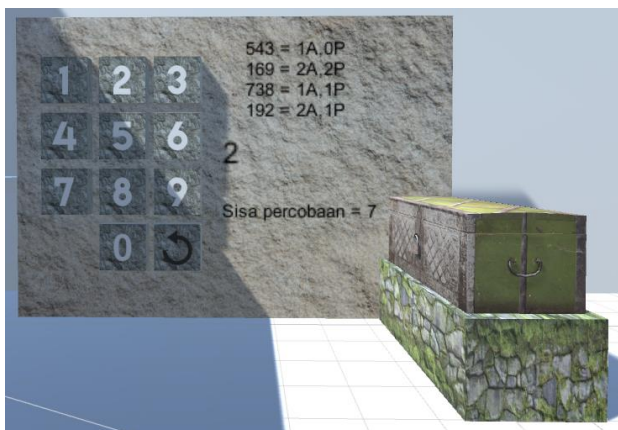
**Tabel 5.3 Skenario Uji Coba *Minigame* Legot**

ID	UF-002
Nama	Uji coba pada <i>minigame</i> Legot.
Tujuan uji coba	Pengguna mengetahui fungsionalitas pada <i>minigame</i> Legot.
Kondisi awal	Pemain berada di depan <i>minigame</i> Legot, peti dalam keadaan tertutup, dan jawaban masih kosong.

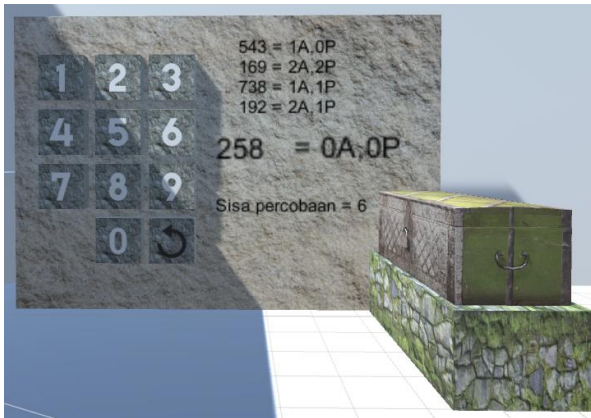
<b><i>Skenario 1</i></b>	<b><i>Minigame mengambil soal dari file .xml.</i></b>
Masukkan	File xml.
Keluaran yang diharapkan	Soal dimunculkan pada <i>minigame</i> .
Hasil uji coba	Berhasil.
Kondisi akhir	<i>Minigame</i> berhasil menampilkan soal.
<b><i>Skenario 2</i></b>	<b><i>Pemain menekan salah satu tombol pada minigame.</i></b>
Masukkan	Pemain mengarahkan kontroler pada tombol angka lalu menekan trigger pada oculus touch.
Keluaran yang diharapkan	Angka yang dipilih ditampilkan pada bagian JawabanPlayer.
Hasil uji coba	Berhasil.
Kondisi akhir	<i>Minigame</i> dapat menampilkan angka yang sesuai dengan tombol yang ditekan.
<b><i>Skenario 3</i></b>	<b><i>Pemain memasukkan jawaban.</i></b>
Masukkan	Pemain menjawab pertanyaan yang disediakan.
Keluaran yang diharapkan	Status kebenaran ditampilkan.
Hasil uji coba	Berhasil.
Kondisi akhir	<i>Minigame</i> dapat menampilkan status kebenaran dengan tepat.
<b><i>Skenario 4</i></b>	<b><i>Pemain menjawab pertanyaan dengan benar.</i></b>
Masukkan	Pemain menjawab pertanyaan dengan benar.
Keluaran yang diharapkan	Peti terbuka.
Hasil uji coba	Berhasil.
Kondisi akhir	Peti dalam keadaan terbuka



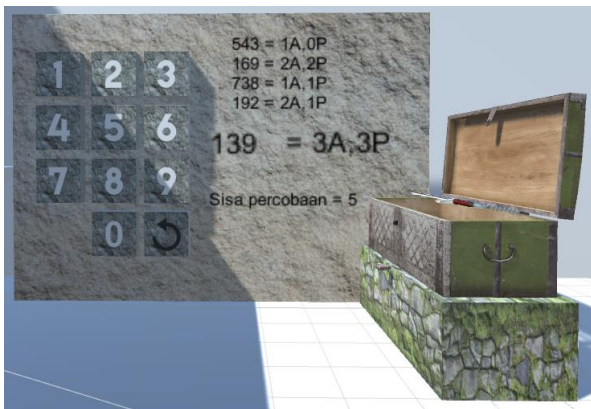
**Gambar 5.3 Hasil uji coba UF-002 Skenario 1**



**Gambar 5.4 Hasil uji coba UF-002 Skenario 2**



**Gambar 5.5 Hasil uji coba UF-002 Skenario 3**



**Gambar 5.6 Hasil uji coba UF-002 Skenario 4**

Hasil dari skenario 1 adalah memunculkan pertanyaan pada *minigame* dapat dilihat pada Gambar 5.3. Berikutnya hasil dari skenario yang memunculkan angka sesuai dengan urutannya bisa dilihat pada Gambar 5.4. Kemudian hasil skenario 3 adalah menghitung status kebenaran dapat dilihat pada Gambar 5.5. Pada skenario 4 dihasilkan peti yang terbuka ketika jawaban sudah benar dapat dilihat pada Gambar 5.6



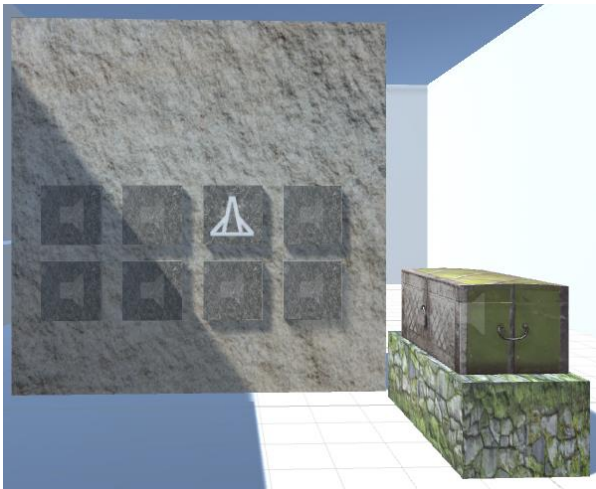
#### 5.1.4 Uji Coba *Minigame* Picture Memory

Pada pengujian *minigame* Picture Memory dilakukan uji coba dengan memainkan *minigame* Picture Memory secara langsung. Skenario yang telah diuji dapat dilihat pada Tabel 5.4.

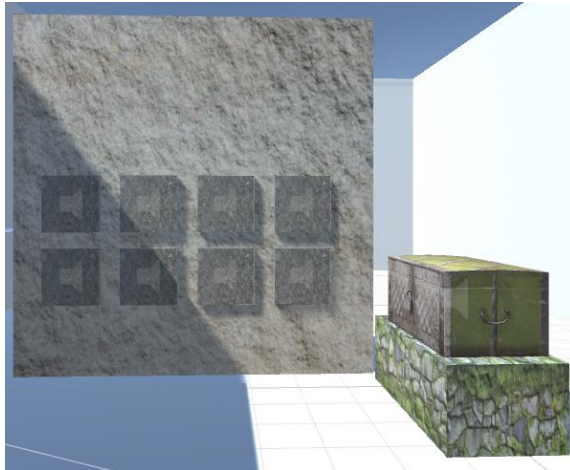
**Tabel 5.4 Skenario Uji Coba *Minigame* Picture Memory**

ID	UF-003
Nama	Uji coba pada <i>minigame</i> Picture memory.
Tujuan uji coba	Pengguna mengetahui fungsionalitas pada <i>minigame</i> Picture memory.
Kondisi awal	Pemain berada di depan <i>minigame</i> Picture Memory, panel masih dalam keadaan membelakangi pemain, dan peti dalam keadaan tertutup.
<b><i>Skenario 1</i></b>	<b><i>Pemain menekan salah satu panel</i></b>
Masukkan	Pemain mengarahkan kontroler pada panel lalu menekan trigger pada oculus touch.
Keluaran yang diharapkan	Panel berputar.
Hasil uji coba	Behasil.
Kondisi akhir	Panel yang dipilih berhasil berputas.
<b><i>Skenario 2</i></b>	<b><i>Pemain memilih dua gambar yang berbeda.</i></b>
Masukkan	Pemain memilih dua panel yang memiliki gambar yang berbeda.
Keluaran yang diharapkan	Panel akan kembali pada posisi terbalik.
Hasil uji coba	Berhasil.
Kondisi akhir	Panel berada pada posisi terbalik.
<b><i>Skenario 3</i></b>	<b><i>Pemain memilih dua gambar yang sama.</i></b>
Masukkan	Pemain memilih dua panel yang memiliki gambar yang sama.

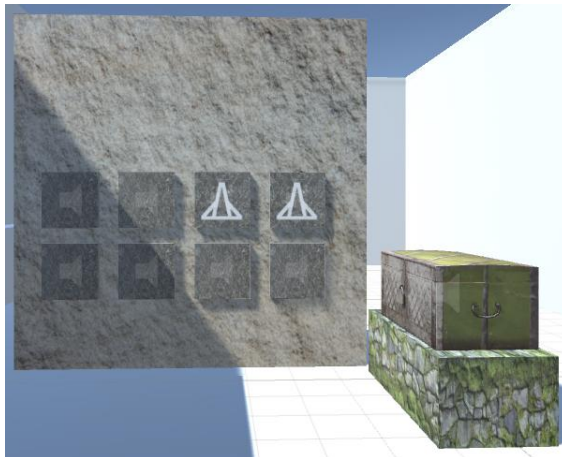
Keluaran yang diharapkan	Panel akan diam pada posisi gambar menghadap pemain.
Hasil uji coba	Berhasil.
Kondisi akhir	Panel berada dalam posisi gambar menghadap pemain.
<b><i>Skenario 4</i></b>	<b><i>Pemain berhasil mencocokkan semua gambar.</i></b>
Masukkan	Pemain mencocokkan semua gambar dengan benar.
Keluaran yang diharapkan	Peti terbuka.
Hasil uji coba	Berhasil.
Kondisi akhir	Peti dalam keadaan terbuka.



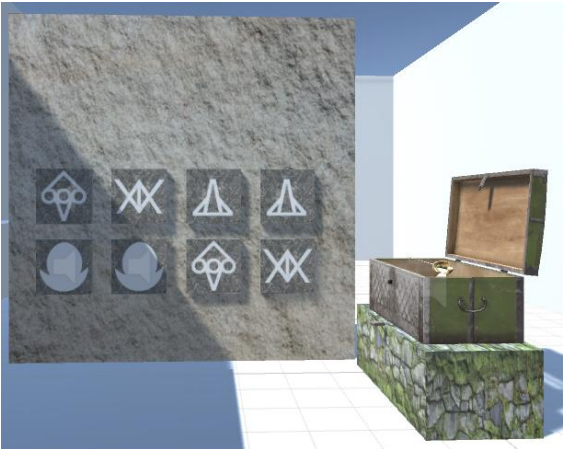
**Gambar 5.7 Hasil uji coba UF-003 Skenario 1**



**Gambar 5.8 Hasil uji coba UF-003 Skenario 2**



**Gambar 5.9 Hasil uji coba UF-003 Skenario 3**



**Gambar 5.10 Hasil uji coba UF-003 Skenario 4**

Hasil uji coba pada skenario 1 menghasilkan panel berputar bisa dilihat pada Gambar 5.7. Lalu untuk skenario 2 menghasilkan panel kembali ke posisi awal dapat dilihat pada Gambar 5.8. Kemudian skenario 3 menghasilkan panel untuk diam pada keadaan gambar menghadap pemain dapat dilihat pada Gambar 5.9. Dan skenario 4 menghasilkan keluaran yaitu peti dalam keadaan terbuka ketika semua gambar sudah menghadap pemain dapat dilihat pada Gambar 5.10.

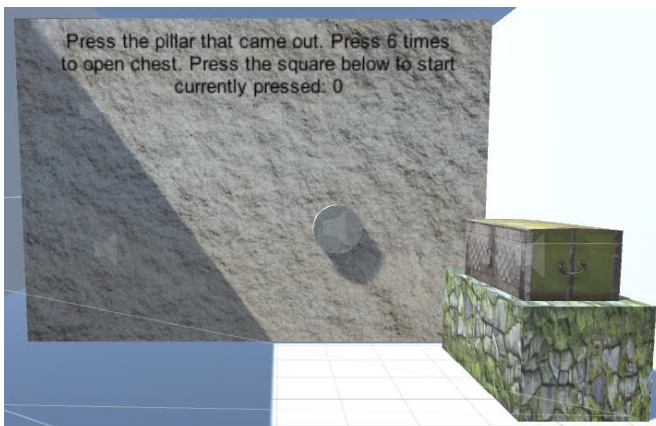
**5.1.5 Uji Coba *Minigame* Tap Fast**

Pada pengujian *minigame* Tap Fast dilakukan uji coba dengan memainkan *minigame* Tap Fast secara langsung. Skenario yang telah diuji dapat dilihat pada Tabel 5.5.

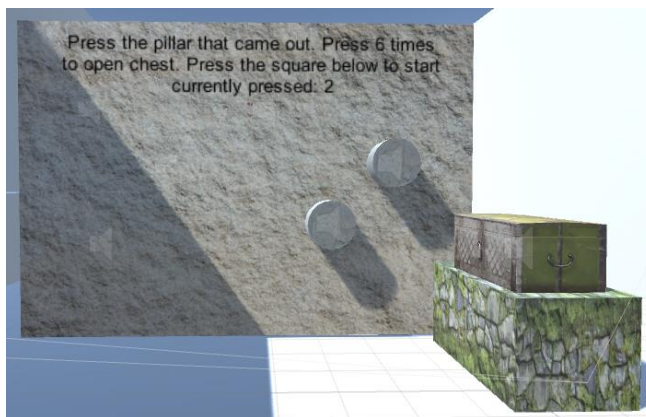
**Tabel 5.5 Skenario Uji Coba *Minigame* Tap Fast**

ID	UF-004
Nama	Uji coba pada <i>minigame</i> Tap Fast.
Tujuan uji coba	Pengguna mengetahui fungsionalitas pada <i>minigame</i> Tap Fast.

Kondisi awal	Pemain berada di depan <i>minigame</i> Tap Fast dan peti dalam keadaan tertutup.
<b><i>Skenario 1</i></b>	<b><i>Pemain menekan tombol Start.</i></b>
Masukkan	Pemain menekan tombol Start yang terletak pada bagian bawah <i>minigame</i> .
Keluaran yang diharapkan	Permainan dimulai dan silinder mulai bergerak keluar dari dalam tembok.
Hasil uji coba	Berhasil.
Kondisi akhir	Permainan berjalan.
<b><i>Skenario 2</i></b>	<b><i>Pemain menekan silinder yang keluar</i></b>
Masukkan	Pemain menekan silinder ketika sedang berada pada posisi di luar tembok.
Keluaran yang diharapkan	Jumlah silinder yang ditekan ditampilkan pada objek TapFastText.
Hasil uji coba	Berhasil.
Kondisi akhir	<i>Minigame</i> menampilkan jumlah silinder yang berhasil ditekan.
<b><i>Skenario 3</i></b>	<b><i>Pemain menekan silinder sejumlah batas minimal untuk menang.</i></b>
Masukkan	Pemain menekan silinder sejumlah dengan yang dibutuhkan untuk menang.
Keluaran yang diharapkan	Peti terbuka
Hasil uji coba	Berhasil.
Kondisi akhir	Peti dalam keadaan terbuka.



**Gambar 5.11 Hasil uji coba UF-004 Skenario 1**



**Gambar 5.12 Hasil uji coba UF-004 Skenario 2**



**Gambar 5.13 Hasil uji coba UF-004 Skenario 3**

Hasil uji coba pada skenario 1 adalah tombol start masuk ke dalam tembok sedangkan silinder mulai keluar dari dalam tembok. Hasil dari skenario 1 dapat dilihat pada Gambar 5.11. Lalu hasil uji coba dari skenario 2 menampilkan jumlah silinder yang telah berhasil ditekan. Bisa dilihat pada Gambar 5.12 bahwa pemain telah menekan sejumlah dua kali. Dan skenario 3 menghasilkan peti untuk terbuka bisa dilihat pada Gambar 5.13.

### 5.1.6 Hasil Pengujian

Dari hasil pengujian yang telah dilakukan maka didapatkan hasil seperti bisa dilihat pada Tabel 5.6

**Tabel 5.6 Hasil Pengujian Fungsionalitas**

ID	Deskripsi	Skenario	Kesimpulan
UF-001	Uji coba kontroler	Skenario 1	Ya
		Skenario 2	Ya
UF-002	Uji coba <i>minigame</i> Legot	Skenario 1	Ya
		Skenario 2	Ya
		Skenario 3	Ya
		Skenario 4	Ya
UF-003		Skenario 1	Ya

	Uji coba <i>minigame</i> Picture memory	Skenario 2	Ya
		Skenario 3	Ya
		Skenario 4	Ya
UF-004	Uji coba <i>minigame</i> Tap Fast	Skenario 1	Ya
		Skenario 2	Ya
		Skenario 3	Ya

## 5.2 Pengujian Pengguna

Selain melakukan uji coba fungsionalitas, dilakukan juga pengujian yang melibatkan pengguna secara langsung. Pengujian ini bernilai subjektif yang bertujuan untuk mengetahui tingkat keberhasilan *minigame* dari sudut pandang pengguna. Penilaian ini bisa didapat dengan meminta penilaian dan tanggapan dari pengguna terhadap aspek-aspek pada *minigame*.

### 5.2.1 Skenario Pengujian Pengguna

Dalam pengujian *minigame*, pengguna diminta untuk mencoba memainkan *minigame* untuk mencoba fungsionalitas dan fitur yang ada. Sebelum pengguna memainkan *minigame* dijelaskan terlebih dahulu cara bermain dan kondisi menang dalam *minigame* tersebut. Setelah itu pengguna diarahkan untuk mencoba memainkan *minigame*. Pada tiap *minigame* terdapat tiga tingkat kesulitan yaitu *easy*, *medium*, dan *hard*. Dalam pengujian, pemain juga melakukan percobaan permainan sebanyak tiga kali, masing-masing mencoba tingkat kesulitan yang berbeda.

Jumlah pengguna yang terlibat dalam pengujian *minigame* sebanyak 7 orang. Pengguna diberikan waktu yang tidak terbatas untuk menyelesaikan tingkat kesulitan *minigame* yang berbeda.

Dalam memberikan penilaian dan tanggapan pengguna diberikan kuesioner pengujian. Pengisian kuesioner dilakukan secara langsung yang diisi sesudah melakukan uji coba. Pada kuisisioner ini diambil beberapa aspek yaitu apakah pengguna sudah familiar dengan lingkungan virtual reality dan perangkat oculus rift yang dapat dilihat pada Tabel 5.7. Lalu diambil juga penilaian dari



masing-masing *minigame* dengan menilai beberapa poin yaitu kesesuaian tingkat kesulitan dan tujuan dari masing-masing *minigame*. Nilai yang diberikan memiliki rentang nilai 1 hingga 6 dengan rentang nilai pada Tabel 5.8. Di akhir pertanyaan mengenai *minigame*, terdapat kolom kritik dan saran untuk *minigame* yang bersangkutan. Pertanyaan untuk menilai *minigame* bisa dilihat pada Tabel 5.9

**Tabel 5.7 Pertanyaan Karakteristik Pengguna**

No.	Karakteristik Pengguna
1.	Apakah anda mengetahui apa itu Virtual Reality?
2.	Apakah anda mengetahui permainan labirin ( <i>maze game</i> )?
3.	Apakah anda pernah bermain permainan labirin?
4.	Apakah anda pernah bermain permainan labirin berbasis <i>virtual reality</i> ?
5.	Jika pernah, bagaimana pendapat anda?
6.	Apakah anda mengetahui <i>Artificial Intelligence</i> ?

**Tabel 5.8 Rentang Nilai**

No.	Keterangan	Nilai
1.	Sangat Tidak Setuju (STS)	1
2.	Tidak Setuju (TS)	2
3.	Kurang Setuju (KS)	3
4.	Cukup Setuju (CS)	4
5.	Setuju (S)	5
6.	Sangat Setuju (SS)	6

**Tabel 5.9 Pertanyaan Penilaian *Minigame***

No	Pernyataan	STS	TS	KS	CS	S	SS
<b>Legot</b>							
1	<i>Minigame</i> Legot dapat melatih kemampuan logika						
2	<i>Minigame</i> Legot pada level <i>easy</i> memiliki						

	tingkat kesulitan yang sesuai						
3	<i>Minigame</i> Legot pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai						
4	<i>Minigame</i> Legot pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai						
Picture Memory							
5	<i>Minigame</i> Picture Memory dapat melatih kemampuan memori						
6	<i>Minigame</i> Picture Memory pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai						
7	<i>Minigame</i> Picture Memory pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai						
8	<i>Minigame</i> Picture Memory pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai						
Tap Fast							
9	<i>Minigame</i> Tap Fast dapat melatih kecepatan refleks						
10	<i>Minigame</i> Tap Fast pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai						

11	<i>Minigame</i> Tap Fast pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai						
12	<i>Minigame</i> Tap Fast pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai						

### 5.2.2 Daftar Penguji

Pada sub bab ini ditunjukkan daftar pengguna yang melakukan pengujian terhadap *minigame*. Untuk menjadi pengguna tidak dibutuhkan kemampuan atau pengalaman khusus dalam hal tertentu karena game ini ditujukan untuk segala kalangan pengguna baik yang senang bermain permainan ataupun tidak. Daftar nama penguji dapat dilihat pada Tabel 5.10.

**Tabel 5.10 Daftar Nama Pengguna**

No.	Nama	Pekerjaan
1.	Ghaly Aditya	Mahasiswa
2.	Riansyah Pramusti	Mahasiswa
3.	Yusuf Dimas Hermawan	Mahasiswa
4.	Rahmat Rijal	Mahasiswa
5.	Sultan Bonar M	Mahasiswa
6.	Ananda Ricky	Mahasiswa
7.	Hari Setiawan	Mahasiswa

### 5.2.3 Hasil Pengujian Pengguna

Sistem penilaian dilakukan menggunakan rentang nilai dari 1 hingga 6 dengan ketentuan nilai 1 sebagai nilai terendah dan nilai 6 sebagai nilai tertinggi. Penilaian akhir diambil dari nilai rata-rata dari masing-masing pertanyaan. Nilai rata-rata diambil dari jumlah nilai lalu dibagi dengan jumlah responden yang menjawab pertanyaan tersebut. Hasil pengujian dan hasil akhir pengujian dapat dilihat pada Tabel 5.11 dan Tabel 5.12.

**Tabel 5.11 Hasil Pengujian Pengguna**

No	Pernyataan	1	2	3	4	5	6
<b>Legot</b>							
1	<i>Minigame</i> Legot dapat melatih kemampuan logika	0	0	0	0	4	3

2	<i>Minigame</i> Legot pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai	0	0	1	0	5	1
3	<i>Minigame</i> Legot pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai	0	0	0	1	4	2
4	<i>Minigame</i> Legot pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai	0	0	0	1	4	2
<b>Picture Memory</b>							
5	<i>Minigame</i> Picture Memory dapat melatih kemampuan memori	0	0	1	0	3	3
6	<i>Minigame</i> Picture Memory pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai	0	0	0	1	3	3
7	<i>Minigame</i> Picture Memory pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai	0	0	0	0	6	1
8	<i>Minigame</i> Picture Memory pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai	0	0	0	1	5	1
<b>Tap Fast</b>							
9	<i>Minigame</i> Tap Fast dapat melatih kecepatan refleks	0	0	1	1	3	2
10	<i>Minigame</i> Tap Fast pada level <i>easy</i>	0	0	1	0	5	1

	memiliki tingkat kesulitan yang sesuai						
11	<i>Minigame</i> Tap Fast pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai	0	0	0	1	6	0
12	<i>Minigame</i> Tap Fast pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai	0	0	0	0	6	1

Tabel 5.12 Hasil Akhir Pengujian Pengguna

No	Pernyataan	Rata-rata	Total	Total (%)
Legot				
1	Minigame Legot dapat melatih kemampuan logika	5,43	5,285	88%
2	Minigame Legot pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai	4,85		
3	Minigame Legot pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai	5,43		
4	Minigame Legot pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai	5,43		
Picture Memory				
5	Minigame Picture Memory dapat melatih kemampuan memori	5,14	5,14	86%

6	Minigame Picture Memory pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai	5,29		
7	Minigame Picture Memory pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai	5,14		
8	Minigame Picture Memory pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai	5		
Tap Fast				
9	Minigame Tap Fast dapat melatih kecepatan refleks	4,85	4,78	80%
10	Minigame Tap Fast pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai	4,29		
11	Minigame Tap Fast pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai	4,85		
12	Minigame Tap Fast pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai	5,14		

#### 5.2.4 Kritik dan Saran Pengguna

Dalam melakukan penilaian pengguna diberikan pertanyaan mengenai kritik dan saran untuk *game* AVIAR secara

keseluruhan. Kritik dan saran pengguna dapat dilihat pada Tabel 5.13.

**Tabel 5.13 Kritik dan Saran Pengguna**

No	Nama	Kritik dan Saran
1.	Ghaly Aditya	Memperlihatkan untuk beberapa detik sebelum permainan dimulai sehingga pemain bisa mengetahui / mengingat jalan yang dituju.
2.	Riansyah Pramusti	Rotasi pemain dibuat halus agar tidak memusingkan (controller).
3.	Yusuf Dimas Hermawan	Dibuat menjadi lebih nyaman untuk dimainkan agar tidak pusing saat bermain.
4.	Rahmat Rijal	Gameplay AVIAR sudah bagus tetapi kalau bisa asset untuk lingkungan gamenya diperbagus lagi agar membuat pemain tertarik dan merasa nyaman.
5.	Sultan Bonar M	Kecepatan saat memakai boots terlalu cepat, bikin pusing. Saran dipelankan (jangan ekstrem).
6.	Ananda Ricky	Gamenya buat pusing, berikan penawar pusing.
7.	Hari Setiawan	Pergerakan pemain terlalu cepat sehingga meninggalkan musuh terlalu jauh. Sound effect kurang menarik.

### 5.3 Evaluasi Pengujian

Pada sub bab ini akan dibahas mengenai evaluasi terhadap hasil pengujian yang telah dilakukan.

Dari hasil uji fungsionalitas pada Tabel 5.6 didapatkan bahwa *minigame* dapat berjalan sesuai dengan skenario dan keluaran yang diharapkan.



Dari data hasil pengujian pengguna pada Tabel 5.11 dan 5.12 didapatkan bahwa masing-masing *minigame* mendapatkan respon baik dari pengguna walaupun masih ada yang harus diperbaiki dan ditambah yang bisa dilihat dari Tabel 5.13 mengenai kritik dan saran.

*(Halaman ini sengaja dikosongkan)*

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan Tugas Akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

#### **6.1 Kesimpulan**

Dari hasil selama proses perancangan, implementasi, dan pengujian dapat diambil kesimpulan sebagai berikut:

1. *Minigame* diimplementasikan dengan menempatkannya pada labirin. Penempatan *minigame* dilakukan dengan membagi labirin menjadi segmen-segmen lalu menempatkan satu *minigame* di dalam masing-masing segmen.
2. Untuk membuat *minigame* yang bermanfaat pertama ditentukan dalam hal apa *minigame* tersebut bisa bermanfaat, yang pada tugas akhir ini adalah kemampuan logika, memori, dan kecepatan refleks.
3. Dalam mendesain tingkat kesulitan bisa dilakukan dengan menambahkan kuantitas dari tantangan pada *minigame*.
4. Kebutuhan fungsionalitas dari kontroler dan semua *minigame* dapat berjalan sesuai dengan yang semestinya.

#### **6.2 Saran**

Berikut saran-saran untuk pengembangan dan perbaikan *minigame* di masa yang akan datang. Diantaranya adalah sebagai berikut:

1. Memperbaiki interaksi kontroler dengan objek agar lebih responsif dan tepat sasaran.
2. Memperindah tampilan *minigame*.
3. Menambahkan jenis *minigame* baru.

*(Halaman ini sengaja dikosongkan)*


## DAFTAR PUSTAKA

- [1] Oculus, "Kickstarter," Juni 2015. [Online]. Available: <https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game>.
- [2] Oculus, "Oculus," 2018. [Online]. Available: <https://www.oculus.com/accessories/>.
- [3] S. Mandal, "Brief Introduction of Virtual Reality & its," *International Journal of Scientific & Engineering Research*, Volume 4, Issue 4, p. 2, 2013.
- [4] S. M. I, Human-Computer Interaction: An Empirical Research Perspective, 2013.
- [5] Unity, "Unity "CREATE THE GAMES YOU LOVE WITH UNITY"," Unity, [Online]. Available: <http://unity3d.com/unity>. [Diakses 27 Mei 2018].
- [6] N. A, "Pemrograman Animasi dan Game Profesional," *Jakarta: Elex Media Komputindo*, 1998.
- [7] Microsoft, "Microsoft studio- development tools," Microsoft, [Online]. Available: <http://www.visualstudio.com>. [Diakses 27 Mei 2018].
- [8] Blender, "About Blender," Blender, [Online]. Available: <http://www.blender.org/about/>. [Diakses 27 Mei 2018].
- [9] "A man who made Whac-A-Mole," *Chibarei*, 2000.
- [10] T. Nelson, "A Brief History of The Master Mind Board Game," Tnelson, 6 August 2014. [Online]. Available: <http://www.tnelson.demon.co.uk/mastermind/history.html>. [Diakses 2018 May 29].

*(Halaman ini sengaja dikosongkan)*

# LAMPIRAN

## A. Hasil Kuesioner


**Kuesioner Tugas Akhir “Game AVIAR”**

5114100065 – Anggit Yudhistira  
 5114100066 – Vinsensia S. Zega  
 5114100124 – Aulfar Rizqi

Surabaya, 09 Juli 2018

---

**IDENTITAS RESPONDEN**

Nama Lengkap : Heri Setiawan  
 Pekerjaan : Mahasiswa  
 Usia : 22 tahun

---

**A. KARAKTERISTIK RESPONDEN**

Isilah pertanyaan di bawah ini dengan menggunakan tanda silang (X)

- Apakah anda mengetahui *Virtual Reality* ?  
☒ Iya, Tahu                      b. Tidak Tahu
- Apakah anda mengetahui permainan labirin (*maze game*) ?  
☒ Iya, Tahu                      b. Tidak Tahu
- Apakah anda pernah bermain permainan labirin ?  
☒ Iya, Pernah                      b. Tidak Pernah
- Apakah anda pernah bermain permainan labirin berbasis *Virtual Reality* ?  
☒ Iya, Pernah                      b. Tidak Pernah  
 Jika Pernah, bagaimana pendapat anda?
- Apakah anda mengetahui *Artificial Intelligence* ?  
☒ Iya, Pernah                      b. Tidak Tahu

---

**B. PENILAIAN TERHADAP APLIKASI**

Isilah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

Keterangan :  
 SS = Sangat setuju                      S = Setuju                      CS = Cukup setuju  
 KS = Kurang setuju                      TS = Tidak Setuju                      STS = Sangat tidak setuju

No	Parameter Antar Muka	STS	TS	KS	CS	S	SS
1	Aplikasi memiliki tampilan, warna, dan desain yang menarik.					✓	
2	Aplikasi ini memiliki tata letak tombol, instruksi dan informasi yang mudah dipahami					✓	
No	Parameter Immersive	STS	TS	KS	CS	S	SS
3	Anda merasakan sensasi nyata seperti di dalam <i>maze</i> sungguhan.						✓
4	Anda merasa tertantang untuk segera mencari jalan keluar <i>maze</i> .						✓
5	Anda merasakan suasana misterius yang ditawarkan oleh permainan				✓		
No	Parameter Artificial Intelligence	STS	TS	KS	CS	S	SS
6	Musuh dalam permainan ini dapat menemukan anda					✓	
7	Semakin tinggi level, maka semakin cepat musuh menemukan anda.					✓	
8	Tingkat kesulitan musuh sesuai dengan level permainan					✓	
No	Parameter Minigame	STS	TS	KS	CS	S	SS
9	Minigame Legot dapat melatih kemampuan logika						✓

	anda						
10	<i>Minigame</i> Legot pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai			✓			
11	<i>Minigame</i> Legot pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
12	<i>Minigame</i> Legot pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
13	<i>Minigame</i> Picture Memory dapat melatih memori anda						✓
14	<i>Minigame</i> Picture Memory pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai						✓
15	<i>Minigame</i> Picture Memory pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai						✓
16	<i>Minigame</i> Picture Memory pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai						✓
17	<i>Minigame</i> Tap Fast dapat melatih kecepatan refleks anda					✓	
18	<i>Minigame</i> Tap Fast pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
19	<i>Minigame</i> Tap Fast pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
20	<i>Minigame</i> Tap Fast pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
No.	Parameter <i>Gameplay</i>	STS	TS	KS	CS	S	SS
21	Rancangan <i>random maze</i> sudah sesuai dengan tingkat kesulitan levelnya						✓
22	Saya merasa <i>game</i> ini cocok dimainkan secara <i>multiplayer</i>					✓	
23	Saya merasa dengan bantuan <i>power up</i> , <i>game</i> ini menjadi menarik						✓
No.	Parameter Kenyamanan	STS	TS	KS	CS	S	SS
24	Aplikasi dapat berjalan lancar tanpa <i>lag</i> dan <i>crash</i>						✓
25	Kontrol pergerakan <i>player</i> tidak membingungkan						✓
26	Saya merasa nyaman selama bermain permainan ini						✓

### C. KRITIK DAN SARAN

.....

.....

.....





## Kuesioner Tugas Akhir "Game AVIAR"

5114100065 – Anggi Yudhistira

5114100066 – Vinsensia S. Zega

5114100124 – Aqur Rizqi

### IDENTITAS RESPONDEN

Nama Lengkap Riansya Pamusti  
Pekerjaan Mahasiswa  
Usia 22

Surabaya, 9 Juli 2018

### A. KARAKTERISTIK RESPONDEN

Isilah pertanyaan di bawah ini dengan menggunakan tanda silang (X)

1. Apakah anda mengetahui *Virtual Reality* ?  
☒ Iya, Tahu      b. Tidak Tahu
2. Apakah anda mengetahui permainan labirin (*maze game*) ?  
☒ Iya, Tahu      b. Tidak Tahu
3. Apakah anda pernah bermain permainan labirin ?  
☒ Iya, Pernah      b. Tidak Pernah
4. Apakah anda pernah bermain permainan labirin berbasis *Virtual Reality* ?  
a. Iya, Pernah      ☒ Tidak Pernah  
Jika Pernah, bagaimana pendapat anda?
5. Apakah anda mengetahui *Artificial Intelligence* ?  
☒ Iya, Pernah      b. Tidak Tahu

### B. PENILAIAN TERHADAP APLIKASI

Isilah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

Keterangan :		S = Setuju		CS = Cukup setuju		STS = Sangat tidak setuju	
SS = Sangat setuju		TS = Tidak Setuju					
KS = Kurang setuju							
No	Parameter Antar Muka	STS	TS	KS	CS	S	SS
1.	Aplikasi memiliki tampilan, warna, dan desain yang menarik					✓	
2.	Aplikasi ini memiliki tata letak tombol, instruksi dan informasi yang mudah dipahami					✓	
No	Parameter Immersive	STS	TS	KS	CS	S	SS
3.	Anda merasakan sensasi nyata seperti di dalam <i>maze</i> sungguhan						✓
4.	Anda merasa tertantang untuk segera mencari jalan keluar <i>maze</i>						✓
5.	Anda merasakan suasana misterius yang ditawarkan oleh permainan					✓	
No	Parameter Artificial Intelligence	STS	TS	KS	CS	S	SS
6.	Musuh dalam permainan ini dapat menemukan anda					✓	
7.	Semakin tinggi level, maka semakin cepat musuh menemukan anda					✓	
8.	Tingkat kesulitan musuh sesuai dengan level permainan				✓		
No	Parameter Minigame	STS	TS	KS	CS	S	SS
9.	Minigame Legot dapat melatih kemampuan logika						✓

	anda							
10	Minigame Legot pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓		
11	Minigame Legot pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai						✓	
12	Minigame Legot pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai						✓	
13	Minigame Picture Memory dapat melatih memori anda						✓	
14	Minigame Picture Memory pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai						✓	
15	Minigame Picture Memory pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓		
16	Minigame Picture Memory pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓		
17	Minigame Tap Fast dapat melatih kecepatan refleks anda					✓		
18	Minigame Tap Fast pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓		
19	Minigame Tap Fast pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓		
20	Minigame Tap Fast pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓		
No	Parameter Gameplay	STS	TS	KS	CS	S	SS	
21	Rancangan <i>random maze</i> sudah sesuai dengan tingkat kesulitan levelnya					✓		
22	Saya merasa <i>game</i> ini cocok dimainkan secara <i>multiplayer</i>							✓
23	Saya merasa dengan bantuan <i>power up</i> , <i>game</i> ini menjadi menarik					✓		
No	Parameter Kenyamanan	STS	TS	KS	CS	S	SS	
24	Aplikasi dapat berjalan lancar tanpa <i>lag</i> dan <i>crash</i>							✓
25	Kontrol pergerakan <i>player</i> tidak membingungkan.					✓		
26	Saya merasa nyaman selama bermain permainan ini					✓		

#### C. KRITIK DAN SARAN

Rotasi pemain dibuat halus agar tidak memusingkan (controller)



## Kuesioner Tugas Akhir "Game AVIAR"

5114100065 – Anggit Yudhistra  
5114100066 – Vinsensia S. Zega  
5114100124 – Aulaz Rizqi

### IDENTITAS RESPONDEN

Nama Lengkap : Yusuf Dhuha H  
Pekerjaan : Mahasiswa  
Usia : 21

Surabaya, 3 Julai 2018

### A. KARAKTERISTIK RESPONDEN

Isilah pertanyaan di bawah ini dengan menggunakan tanda silang (X)

- Apakah anda mengetahui *Virtual Reality* ?  
☒ a. Iya, Tahu ☐ b. Tidak Tahu
- Apakah anda mengetahui permainan labirin (*maze game*) ?  
☒ a. Iya, Tahu ☐ b. Tidak Tahu
- Apakah anda pernah bermain permainan labirin ?  
☒ a. Iya, Pernah ☐ b. Tidak Pernah
- Apakah anda pernah bermain permainan labirin berbasis *Virtual Reality* ?  
☐ a. Iya, Pernah ☒ b. Tidak Pernah  
Jika Pernah, bagaimana pendapat anda?  
.....
- Apakah anda mengetahui *Artificial Intelligence* ?  
☒ a. Iya, Pernah ☐ b. Tidak Tahu

### B. PENILAIAN TERHADAP APLIKASI

Isilah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

Keterangan :							
SS	= Sangat setuju	S	= Setuju	CS	= Cukup setuju		
KS	= Kurang setuju	TS	= Tidak Setuju	STS	= Sangat tidak setuju		
No	Parameter Antar Muka	STS	TS	KS	CS	S	SS
1.	Aplikasi memiliki tampilan, warna, dan desain yang menarik.					✓	
2.	Aplikasi ini memiliki tata letak tombol, instruksi dan informasi yang mudah dipahami				✓		
No	Parameter <i>Immersive</i>	STS	TS	KS	CS	S	SS
3.	Anda merasakan sensasi nyata seperti di dalam <i>maze</i> sungguhan.						✓
4.	Anda merasa tertantang untuk segera mencari jalan keluar <i>maze</i>					✓	
5.	Anda merasakan suasana misterius yang ditawarkan oleh permainan				✓		
No	Parameter <i>Artificial Intelligence</i>	STS	TS	KS	CS	S	SS
6.	Musuh dalam permainan ini dapat menemukan anda					✓	
7.	Semakin tinggi level, maka semakin cepat musuh menemukan anda				✓		
8.	Tingkat kesulitan musuh sesuai dengan level permainan					✓	
No	Parameter <i>Minigame</i>	STS	TS	KS	CS	S	SS
9.	<i>Minigame</i> Legot dapat melatih kemampuan logika					✓	

	anda						
10	Minigame Legot pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
11	Minigame Legot pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
12	Minigame Legot pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
13	Minigame Picture Memory dapat melatih memori anda					✓	
14	Minigame Picture Memory pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
15	Minigame Picture Memory pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
16	Minigame Picture Memory pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
17	Minigame Tap Fast dapat melatih kecepatan refleks anda						✓
18	Minigame Tap Fast pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
19	Minigame Tap Fast pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
20	Minigame Tap Fast pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
No	Parameter <i>Gameplay</i>	STS	TS	KS	CS	S	SS
21	Rancangan <i>random maze</i> sudah sesuai dengan tingkat kesulitan levelnya						✓
22	Saya merasa <i>game</i> ini cocok dimainkan secara <i>multiplayer</i>				✓		
23	Saya merasa dengan bantuan <i>power up</i> , <i>game</i> ini menjadi menarik					✓	
No	Parameter Kenyamanan	STS	TS	KS	CS	S	SS
24	Aplikasi dapat berjalan lancar tanpa <i>lag</i> dan <i>crash</i>					✓	
25	Kontrol pergerakan <i>player</i> tidak membingungkan.					✓	
26	Saya merasa nyaman selama bermain permainan ini				✓		

#### C. KRITIK DAN SARAN

Ditelaah menjadi lebih nyaman untuk dimainkan agar tidak pusing saat bermain.



ITS  
Institut Teknologi Sepuluh Nopember

## Kuesioner Tugas Akhir "Game AVIAR"

5114100065 - Anggit Yudhisura  
5114100066 - Vinsensia S. Zega  
5114100124 - Auffer Rizqi

### IDENTITAS RESPONDEN

Nama Lengkap : Ethan Bonay M.  
Pekerjaan : Mahasiswa  
Usia : 21

Surabaya, ..... 2018

### A. KARAKTERISTIK RESPONDEN

Isilah pertanyaan di bawah ini dengan menggunakan tanda silang (X)

- Apakah anda mengetahui *Virtual Reality* ?  
☒ Ya, Tahu      b. Tidak Tahu
- Apakah anda mengetahui permainan labirin (*maze game*) ?  
☒ Ya, Tahu      b. Tidak Tahu
- Apakah anda pernah bermain permainan labirin ?  
☒ Ya, Pernah      b. Tidak Pernah
- Apakah anda pernah bermain permainan labirin berbasis *Virtual Reality* ?  
 a. Iya, Pernah      ☒ Tidak Pernah  
 Jika Pernah, bagaimana pendapat anda?  
 .....
- Apakah anda mengetahui *Artificial Intelligence* ?  
☒ Ya, Pernah      b. Tidak Tahu

### B. PENILAIAN TERHADAP APLIKASI

Isilah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

Keterangan :							
SS = Sangat setuju		S = Setuju		CS = Cukup setuju			
KS = Kurang setuju		TS = Tidak Setuju		STS = Sangat tidak setuju			
No	Parameter Antar Muka	STS	TS	KS	CS	S	SS
1.	Aplikasi memiliki tampilan, warna, dan desain yang menarik					✓	
2.	Aplikasi ini memiliki tata letak tombol, instruksi dan informasi yang mudah dipahami					✓	
No	Parameter Immersive	STS	TS	KS	CS	S	SS
3.	Anda merasakan sensasi nyata seperti di dalam <i>maze</i> sungguhan					✓	
4.	Anda merasa tertantang untuk segera mencari jalan keluar <i>maze</i>					✓	
5.	Anda merasakan suasana misterius yang ditawarkan oleh permainan					✓	
No	Parameter Artificial Intelligence	STS	TS	KS	CS	S	SS
6.	Musuh dalam permainan ini dapat menemukan anda				✓		
7.	Semakin tinggi level, maka semakin cepat musuh menemukan anda				✓		
8.	Tingkat kesulitan musuh sesuai dengan level permainan				✓		
No	Parameter Minigame	STS	TS	KS	CS	S	SS
9.	Minigame Legot dapat melatih kemampuan logika					✓	

	anda						
10	Minigame Legot pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai						✓
11	Minigame Legot pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
12	Minigame Legot pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
13	Minigame Picture Memory dapat melatih memori anda					✓	
14	Minigame Picture Memory pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
15	Minigame Picture Memory pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
16	Minigame Picture Memory pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
17	Minigame Tap Fast dapat melatih kecepatan refleks anda				✓		
18	Minigame Tap Fast pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
19	Minigame Tap Fast pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
20	Minigame Tap Fast pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
No.	Parameter Gameplay	STS	TS	KS	CS	S	SS
21	Rancangan <i>random maze</i> sudah sesuai dengan tingkat kesulitan levelnya					✓	
22	Saya merasa <i>game</i> ini cocok dimainkan secara <i>multiplayer</i>					✓	
23	Saya merasa dengan bantuan <i>power up</i> , <i>game</i> ini menjadi menarik					✓	
No.	Parameter Kenyamanan	STS	TS	KS	CS	S	SS
24	Aplikasi dapat berjalan lancar tanpa <i>lag</i> dan <i>crash</i>						✓
25	Kontrol pergerakan <i>player</i> tidak membingungkan				✓		
26	Saya merasa nyaman selama bermain permainan ini						

### C. KRITIK DAN SARAN

Kepuasan saat memakai boots terlalu cepat bikin  
 tangan dipelankan (jangan ekstrem)





## Kuesioner Tugas Akhir "Game AVIAR"

5114100065 - Anggit Yudhistira  
5114100066 - Vinsensia S. Zega  
5114100124 - Aulur Rizqi

### IDENTITAS RESPONDEN

Nama Lengkap : Ghalq Aditya  
Pekerjaan : Mahasiswa  
Usia : 19

Surabaya, 9 Juli 2018

*[Signature]*

### A. KARAKTERISTIK RESPONDEN

Isilah pertanyaan di bawah ini dengan menggunakan tanda silang (X)

- Apakah anda mengetahui *Virtual Reality* ?  
☒ a. Iya, Tahu      b. Tidak Tahu
- Apakah anda mengetahui permainan labirin (*maze game*) ?  
☒ a. Iya, Tahu      b. Tidak Tahu
- Apakah anda pernah bermain permainan labirin ?  
a. Iya, Pernah      ☒ b. Tidak Pernah
- Apakah anda pernah bermain permainan labirin berbasis *Virtual Reality* ?  
a. Iya, Pernah      ☒ b. Tidak Pernah  
Jika Pernah, bagaimana pendapat anda?  
.....
- Apakah anda mengetahui *Artificial Intelligence* ?  
☒ a. Iya, Pernah      b. Tidak Tahu

### B. PENILAIAN TERHADAP APLIKASI

Isilah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

Keterangan		S = Setuju		CS = Cukup setuju		TS = Tidak Setuju		STS = Sangat tidak setuju	
SS = Sangat setuju		KS = Kurang setuju							
No	Parameter Antar Muka	STS	TS	KS	CS	S	SS		
1.	Aplikasi memiliki tampilan, warna, dan desain yang menarik.			✓					
2.	Aplikasi ini memiliki tata letak tombol, instruksi dan informasi yang mudah dipahami								✓
No	Parameter Immersive	STS	TS	KS	CS	S	SS		
3.	Anda merasakan sensasi nyata seperti di dalam <i>maze</i> sungguhan				✓				
4.	Anda merasa tertantang untuk segera mencari jalan keluar <i>maze</i>					✓			
5.	Anda merasakan suasana misterius yang ditawarkan oleh permainan			✓					
No	Parameter Artificial Intelligence	STS	TS	KS	CS	S	SS		
6.	Musuh dalam permainan ini dapat menemukan anda					✓			
7.	Semakin tinggi level, maka semakin cepat musuh menemukan anda								✓
8.	Tingkat kesulitan musuh sesuai dengan level permainan				✓				
No	Parameter Minigame	STS	TS	KS	CS	S	SS		
9.	<i>Minigame</i> Legot dapat melatih kemampuan logika					✓			

	anda						
10	Minigame Legot pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
11	Minigame Legot pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai				✓		
12	Minigame Legot pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai				✓		
13	Minigame Picture Memory dapat melatih memori anda			✓			
14	Minigame Picture Memory pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai				✓		
15	Minigame Picture Memory pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
16	Minigame Picture Memory pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai				✓		
17	Minigame Tap Fast dapat melatih kecepatan refleks anda			✓			
18	Minigame Tap Fast pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai			✓			
19	Minigame Tap Fast pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai				✓		
20	Minigame Tap Fast pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
No	Parameter Gameplay	STS	TS	KS	CS	S	SS
21	Rancangan <i>random maze</i> sudah sesuai dengan tingkat kesulitan levelnya				✓		
22	Saya merasa <i>game</i> ini cocok dimainkan secara <i>multiplayer</i>		✓				
23	Saya merasa dengan bantuan <i>power up</i> , <i>game</i> ini menjadi menarik					✓	
No	Parameter Kenyamanan	STS	TS	KS	CS	S	SS
24	Aplikasi dapat berjalan lancar tanpa <i>lag</i> dan <i>crash</i>				✓		
25	Kontrol pergerakan <i>player</i> tidak membingungkan						✓
26	Saya merasa nyaman selama bermain permainan ini			✓			

### C. KRITIK DAN SARAN

Menperlihatkan utk bbrp detik sebelum permainan dimulai  
 Sehingga pemain bisa mengetahui mengenai jalan yg dituju





ITS  
Institut Teknologi Sepuluh Nopember

## Kuesioner Tugas Akhir "Game AVIAR"

5114100065 – Anggit Yudhistira

5114100066 – Vinsensia S. Zega

5114100124 – Afiar Rizqi

### IDENTITAS RESPONDEN

Nama Lengkap : Ananda Kicker  
Pekerjaan : Malaysia Gen 8  
Usia : 22 tahun

Surabaya, 9 Juli 2018

### A. KARAKTERISTIK RESPONDEN

Isilah pertanyaan di bawah ini dengan menggunakan tanda silang (X)

- Apakah anda mengetahui *Virtual Reality* ?  
☒ Iya, Tahu      b. Tidak Tahu
  - Apakah anda mengetahui permainan labirin (*maze game*) ?  
☒ Iya, Tahu      b. Tidak Tahu
  - Apakah anda pernah bermain permainan labirin ?  
☒ Iya, Pernah      b. Tidak Pernah
  - Apakah anda pernah bermain permainan labirin berbasis *Virtual Reality* ?  
☒ Iya, Pernah      b. Tidak Pernah
- Jika Pernah, bagaimana pendapat anda?  
Ya, tapi ngga? Sehep
- Apakah anda mengetahui *Artificial Intelligence* ?  
☒ Iya, Pernah      b. Tidak Tahu

### B. PENILAIAN TERHADAP APLIKASI

Isilah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

Keterangan		SS = Sangat setuju	S = Setuju	CS = Cukup setuju	STS = Sangat tidak setuju	TS = Tidak Setuju	KS = Kurang setuju	SS
No	Parameter Antar Muka	STS	TS	KS	CS	S	SS	
1.	Aplikasi memiliki tampilan, warna, dan desain yang menarik.							✓
2.	Aplikasi ini memiliki tata letak tombol, instruksi dan informasi yang mudah dipahami				✓	✓		
No	Parameter Immersive	STS	TS	KS	CS	S	SS	
3.	Anda merasakan sensasi nyata seperti di dalam <i>maze</i> sungguhan					✓		
4.	Anda merasa tertantang untuk segera mencari jalan keluar <i>maze</i>				✓			
5.	Anda merasakan suasana misterius yang ditawarkan oleh permainan				✓			
No	Parameter Artificial Intelligence	STS	TS	KS	CS	S	SS	
6.	Musuh dalam permainan ini dapat menemukan anda			✓				
7.	Semakin tinggi level, maka semakin cepat musuh menemukan anda			✓				
8.	Tingkat kesulitan musuh sesuai dengan level permainan		✓					
No	Parameter Minigame	STS	TS	KS	CS	S	SS	
9.	<i>Minigame</i> Legot dapat melatih kemampuan logika					✓		

	anda						
10	Minigame Legot pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
11	Minigame Legot pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
12	Minigame Legot pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
13	Minigame Picture Memory dapat melatih memori anda					✓	
14	Minigame Picture Memory pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
15	Minigame Picture Memory pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
16	Minigame Picture Memory pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
17	Minigame Tap Fast dapat melatih kecepatan refleks anda					✓	
18	Minigame Tap Fast pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
19	Minigame Tap Fast pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
20	Minigame Tap Fast pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
No.	Parameter Gameplay	STS	TS	KS	CS	S	SS
21	Rancangan <i>random maze</i> sudah sesuai dengan tingkat kesulitan levelnya					✓	
22	Saya merasa <i>game</i> ini cocok dimainkan secara <i>multiplayer</i>						✓
23	Saya merasa dengan bantuan <i>power up</i> , <i>game</i> ini menjadi menarik				✓		
No.	Parameter Kenyamanan	STS	TS	KS	CS	S	SS
24	Aplikasi dapat berjalan lancar tanpa <i>lag</i> dan <i>crash</i>					✓	
25	Kontrol pergerakan <i>player</i> tidak membingungkan.		✓	✓			
26	Saya merasa nyaman selama bermain permainan ini						

#### C. KRITIK DAN SARAN

Gamernya *gampang* banget *pusing*, bikin *permainan* *pusing*.



## Kuesioner Tugas Akhir "Game AVIAR"

5114100065 – Anggi Yudhistira

5114100066 – Vinsensia S. Zega

5114100124 – Aulfar Rizqi

### IDENTITAS RESPONDEN

Nama Lengkap : Rahmat Rizal  
Pekerjaan : Mahasiswa  
Usia : 22 tahun

Surabaya, 22 Desember 2018

Rahmat Rizal

### A. KARAKTERISTIK RESPONDEN

Isilah pertanyaan di bawah ini dengan menggunakan tanda silang (X)

1. Apakah anda mengetahui *Virtual Reality* ?  
☒ Iya, Tahu      b. Tidak Tahu
2. Apakah anda mengetahui permainan labirin (*maze game*) ?  
☒ Iya, Tahu      b. Tidak Tahu
3. Apakah anda pernah bermain permainan labirin ?  
☒ Iya, Pernah      b. Tidak Pernah
4. Apakah anda pernah bermain permainan labirin berbasis *Virtual Reality* ?  
a. Iya, Pernah      ☒ Tidak Pernah  
Jika Pernah, bagaimana pendapat anda?
5. Apakah anda mengetahui *Artificial Intelligence* ?  
☒ Iya, Pernah      b. Tidak Tahu

### B. PENILAIAN TERHADAP APLIKASI

Isilah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

Keterangan :							
SS = Sangat setuju		S = Setuju		CS = Cukup setuju			
KS = Kurang setuju		TS = Tidak Setuju		STS = Sangat tidak setuju			
No	Parameter Antar Muka	STS	TS	KS	CS	S	SS
1.	Aplikasi memiliki tampilan, warna, dan desain yang menarik.						✓
2.	Aplikasi ini memiliki tata letak tombol, instruksi dan informasi yang mudah dipahami						✓
No	Parameter Immersive	STS	TS	KS	CS	S	SS
3.	Anda merasakan sensasi nyata seperti di dalam <i>maze</i> sungguhan						✓
4.	Anda merasa tertantang untuk segera mencari jalan keluar <i>maze</i>					✓	
5.	Anda merasakan suasana misterius yang ditawarkan oleh permainan					✓	
No	Parameter Artificial Intelligence	STS	TS	KS	CS	S	SS
6.	Musuh dalam permainan ini dapat menemukan anda					✓	
7.	Semakin tinggi level, maka semakin cepat musuh menemukan anda						✓
8.	Tingkat kesulitan musuh sesuai dengan level permainan					✓	
No	Parameter Minigame	STS	TS	KS	CS	S	SS
9.	Minigame Legot dapat melatih kemampuan logika						✓



## BIODATA PENULIS



Aufar Rizqi, lahir di Jakarta 16 Maret 1996. Penulis merupakan anak ke dua dari 2 bersaudara dari pasangan Bapak Ismunanda dan Ibu Nunung Dyah Hapsari. Penulis menempuh pendidikan formal dari TK Bina Insani (2000-2002), SD Madania (2002-2008), SMP Islam Terpadu Ummul Quro (2008-2011, SMA Pesantren Unggul Al Bayan (2011-2014), dan S1 Informatika ITS (2014-2018). Selama mengemban pendidikan perkuliahan penulis mengambil rukun mata kuliah IGS (Interaksi Grafik dan Seni). Penulis juga aktif dalam organisasi mahasiswa HMTC (Himpunan Mahasiswa Teknik Computer-Informatika) sebagai staff Departemen Hubungan Luar. Selain itu penulis juga aktif dalam kegiatan kepanitiaan seperti Schematics 2015 dan 2016, ITS Expo 2015, 2016, dan 2017, dan Gerigi ITS 2015 dan 2016. Penulis memiliki ketertarikan dalam bermain dan membuat game. Komunikasi dengan penulis dapat melalui email: **aufarr@gmail.com**.